
Google™
10
I/O

Casting a Wide Net: Targeting All Android Devices

Justin Mattson
Developer Advocate
19 May 2010

House keeping

Use Wave to see live notes and ask questions

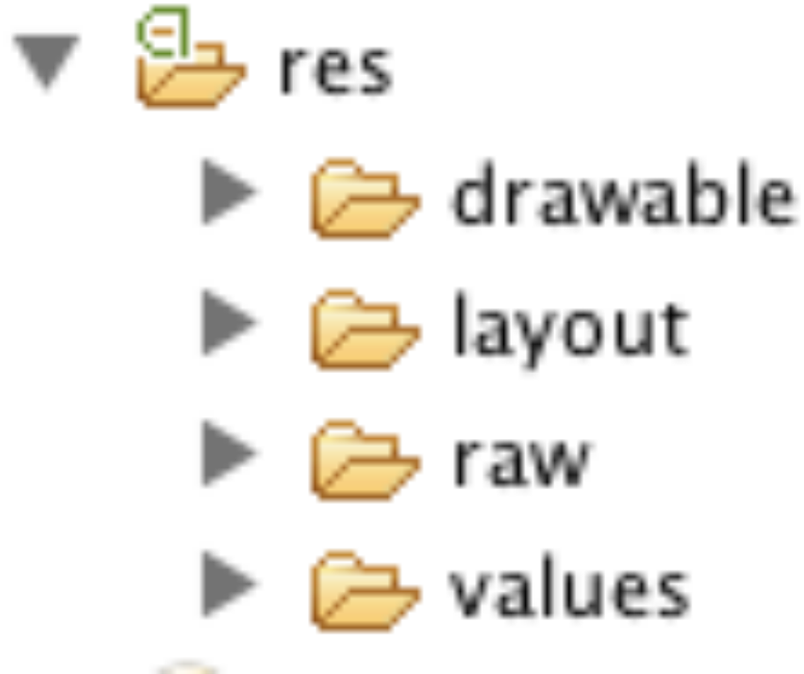
http://tinyurl.com/io10casting

Agenda

- Resource Loading
- Image & Drawable Considerations
- Adapting to API Availability
- Testing

Resource Loading

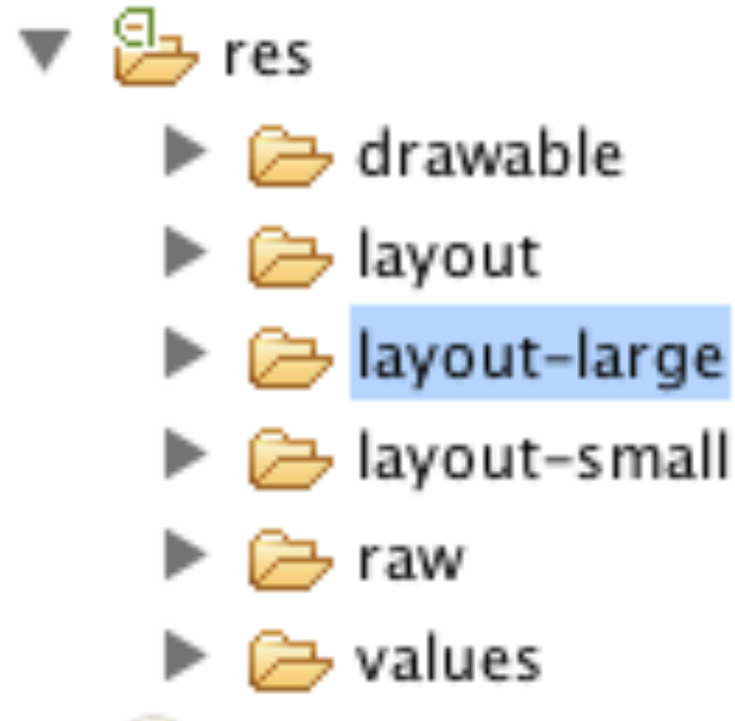
- Based on current system configuration
- If config changes, your app restarts (eg. orientation change)
- Hierarchy of qualifiers



Resource loading

Screen size

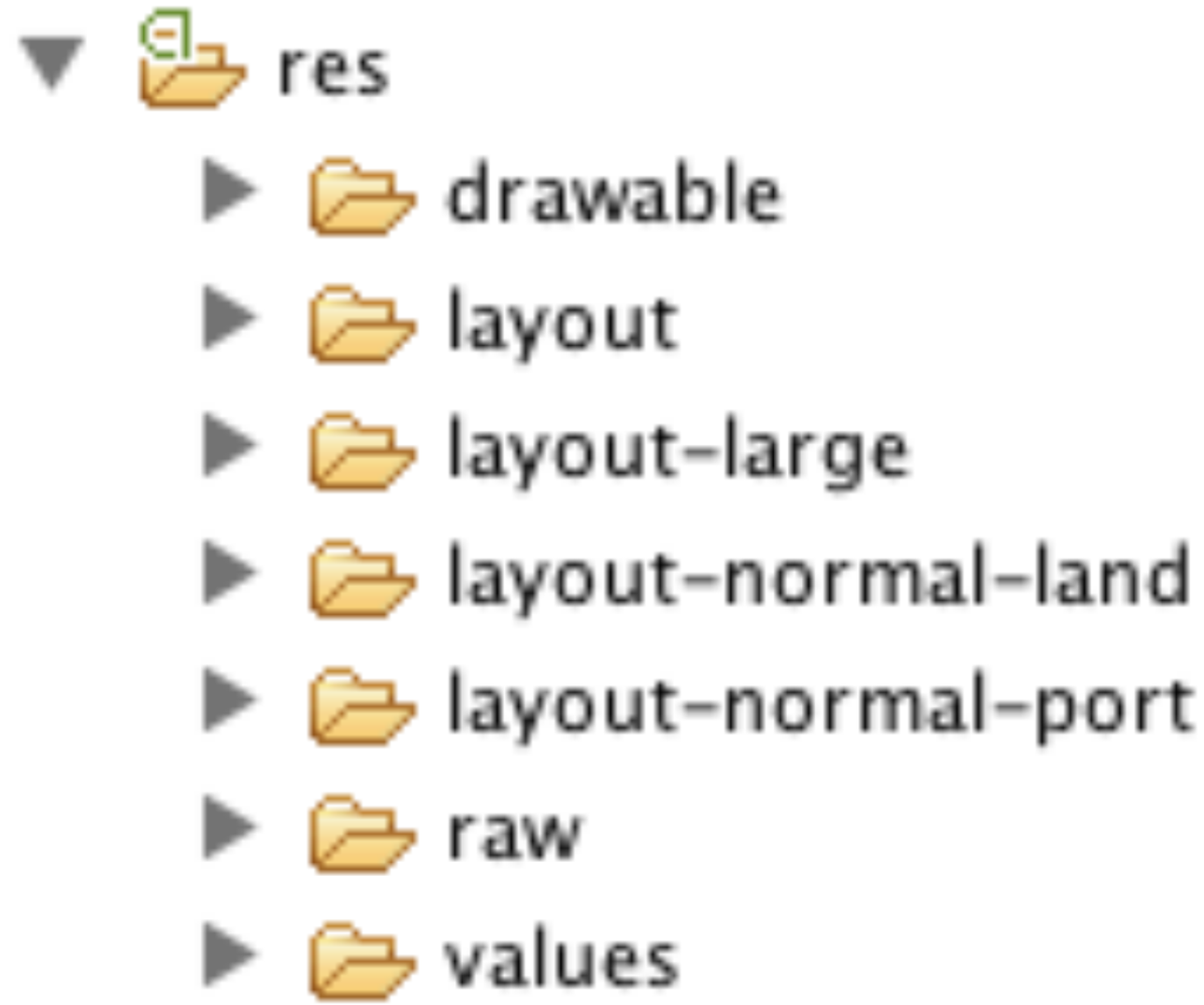
- Physical size of the screen
- Buckets
  - small: <~3.0”
  - normal: ~3.0” - ~4.0”
  - large: >~4.0”



Resource loading

Screen orientation

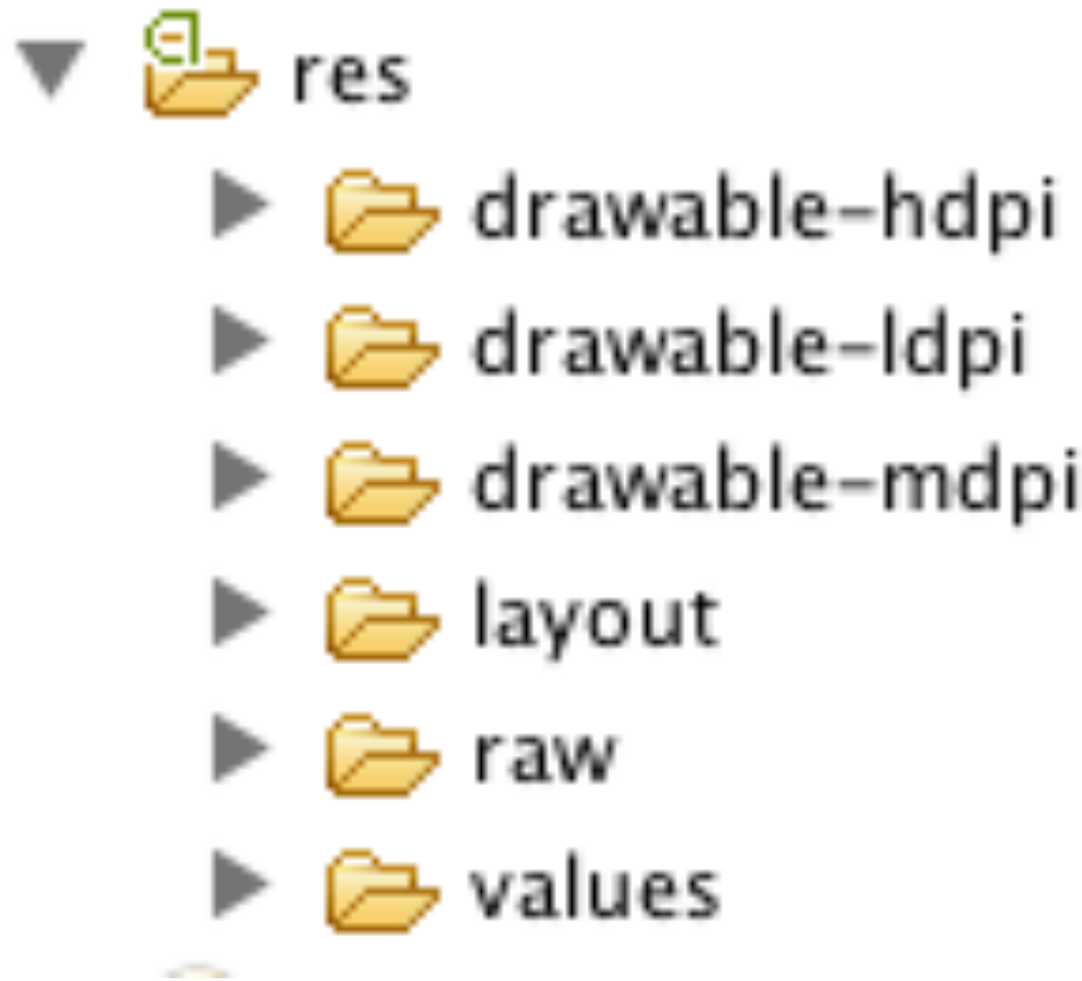
- landscape, portrait, or square
- Might be triggered by accelerometers or other events



Resource loading

Screen density

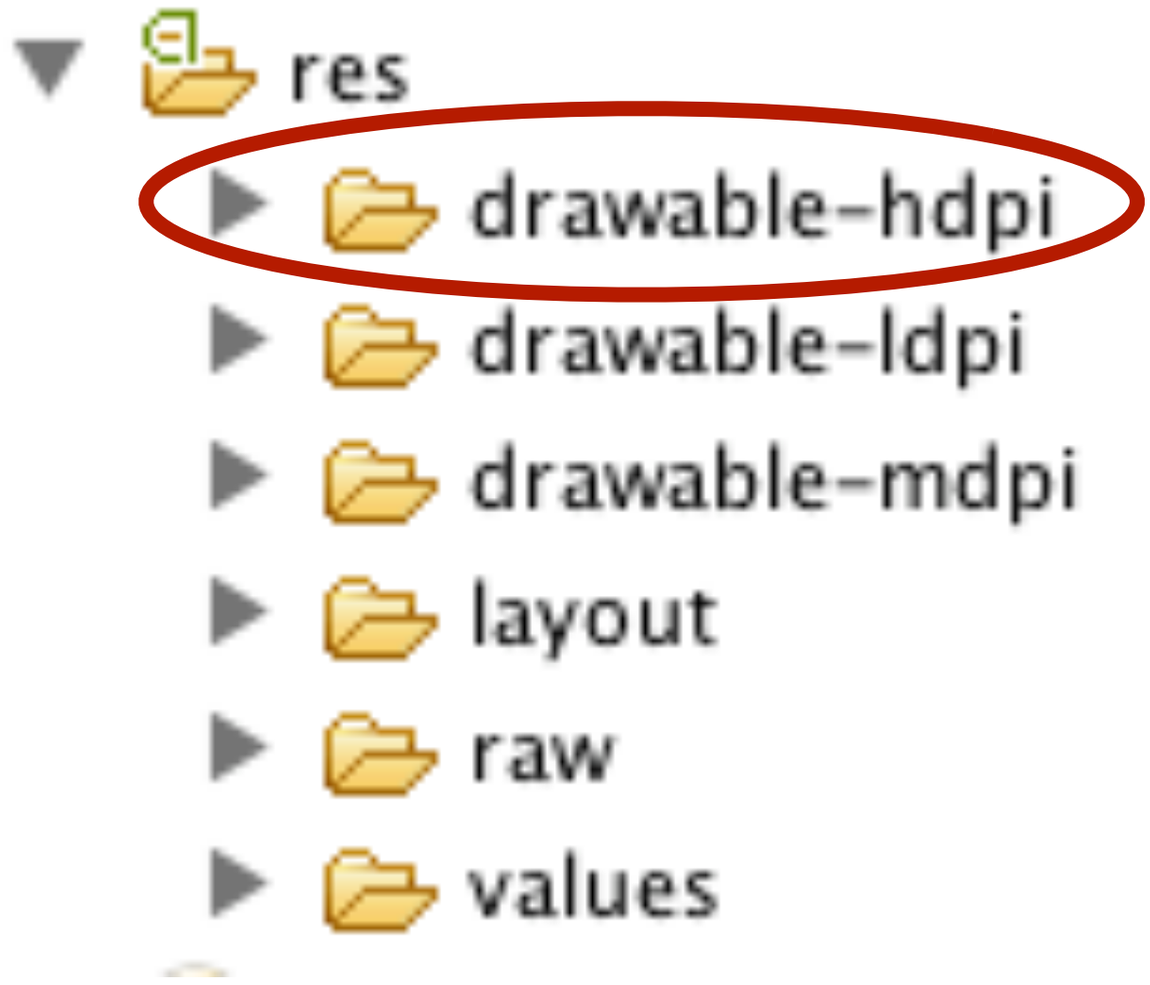
- Pixels per unit of measurement
- System looks for “best” fit*
- Buckets
  - ldpi: ~120dpi
  - mdpi: ~160dpi
  - hdpi: ~240dpi
- *Matching: everyone is a winner
  - first match on 1.5



Resource loading

Screen density

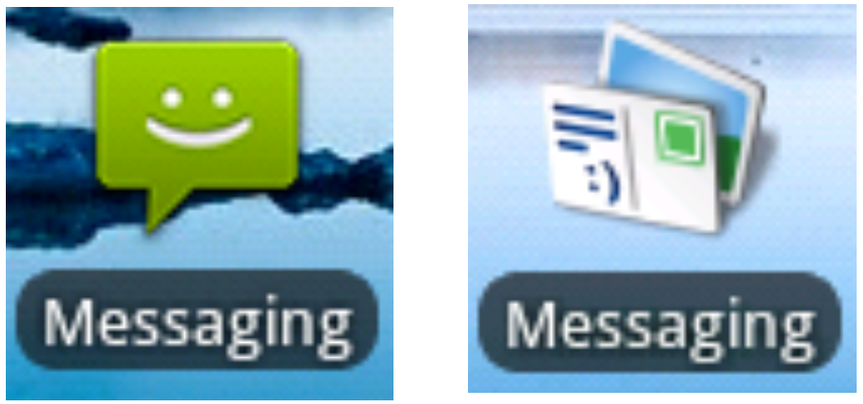
- Pixels per unit of measurement
- System looks for “best” fit*
- Buckets
  - ldpi: ~120dpi
  - mdpi: ~160dpi
  - hdpi: ~240dpi
- *Matching: everyone is a winner
  - first match on 1.5



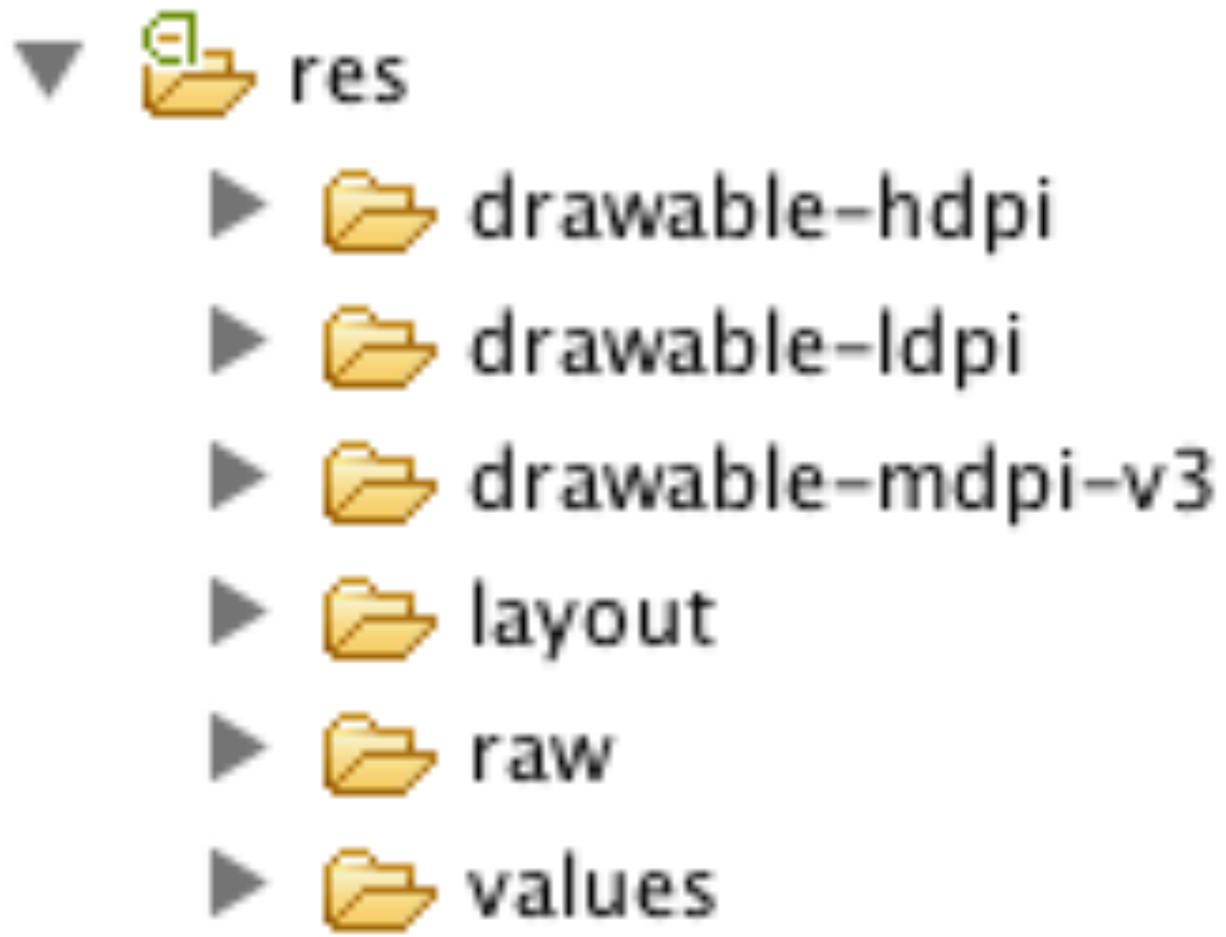
Resource loading

SDK Version

- Styles across versions



- Inexact matching, <= current version

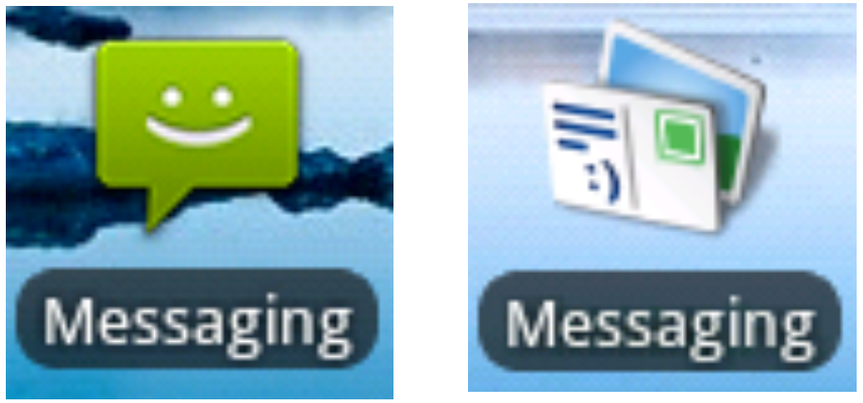


- Work around 1.5 dpi bug
  - Exact matching on 1.5 - 2.0
  - v5 thinks its v6

Resource loading

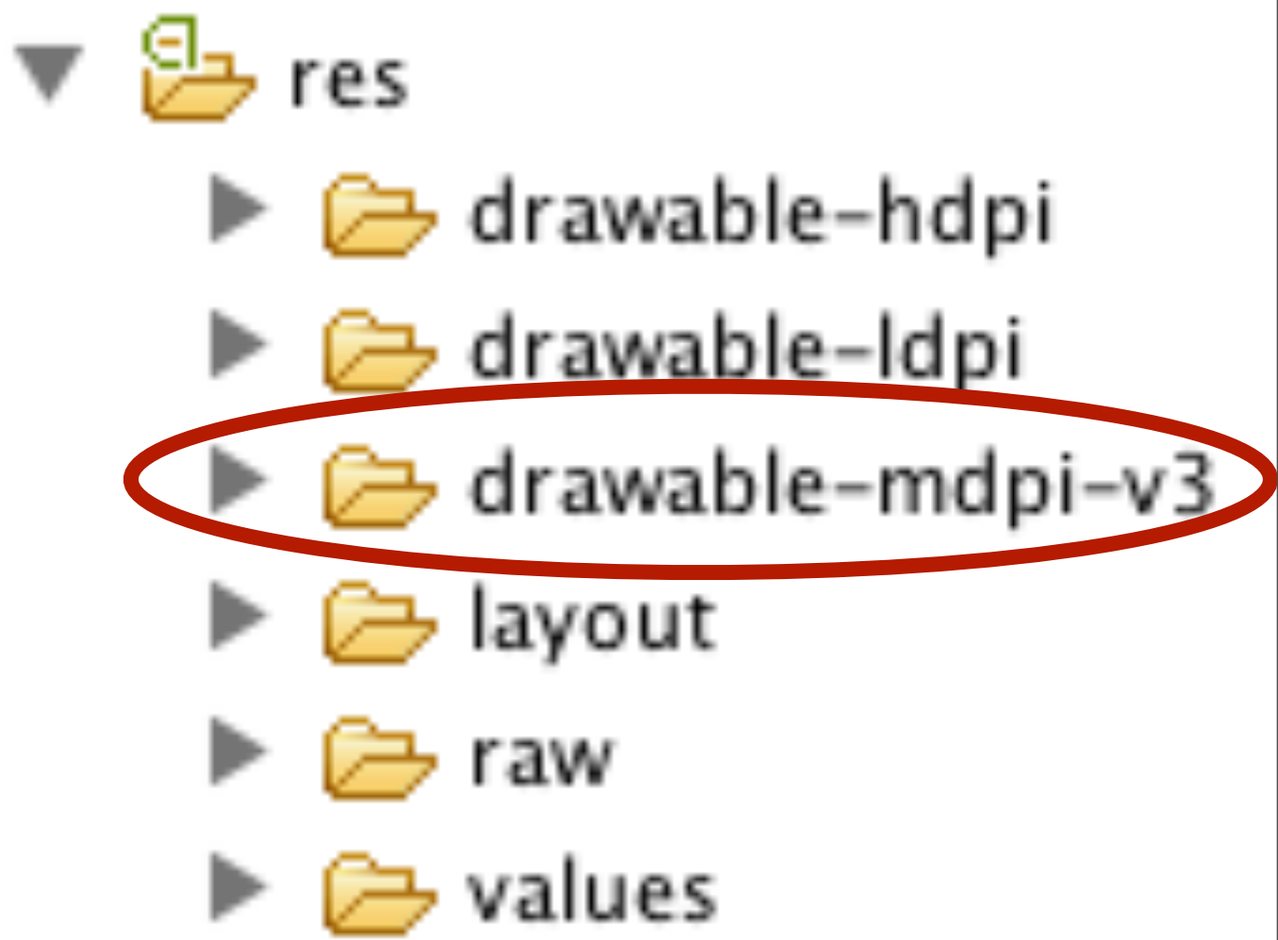
SDK Version

- Styles across versions



- Inexact matching, <= current version

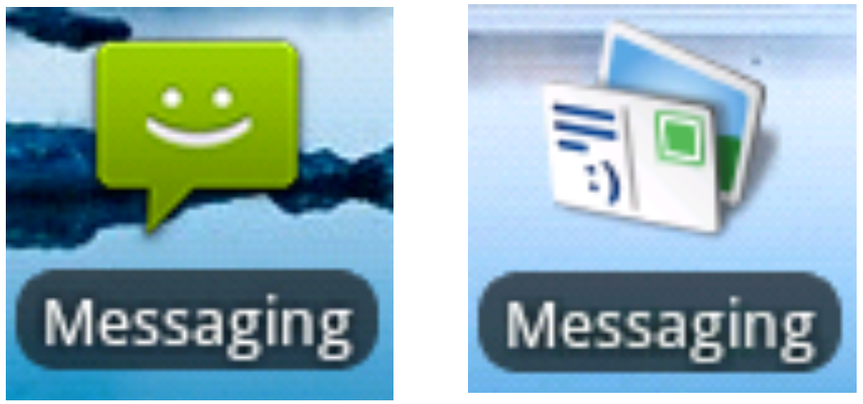
- Work around 1.5 dpi bug
  - Exact matching on 1.5 - 2.0
  - v5 thinks its v6



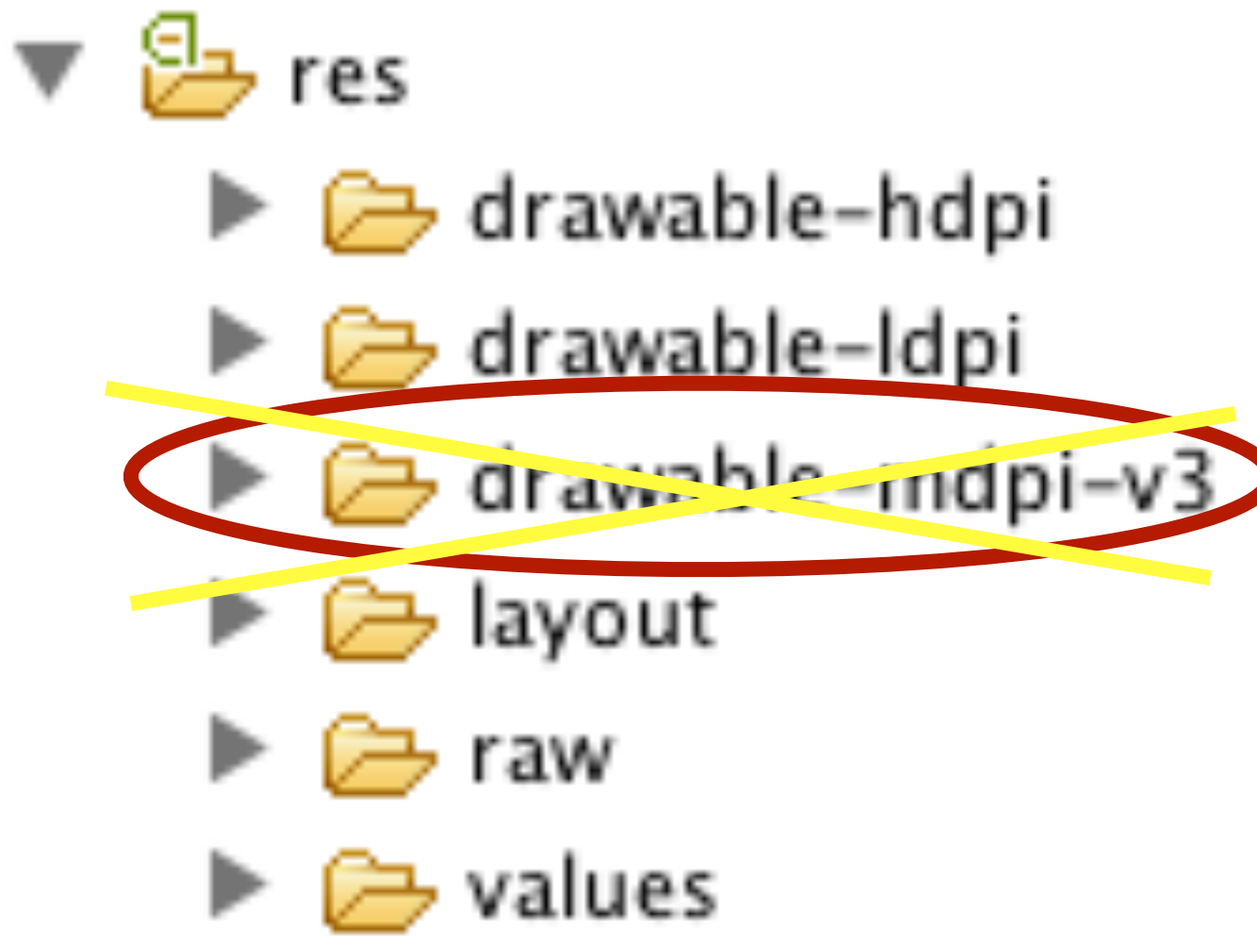
Resource loading

SDK Version

- Styles across versions



- Inexact matching, <= current version

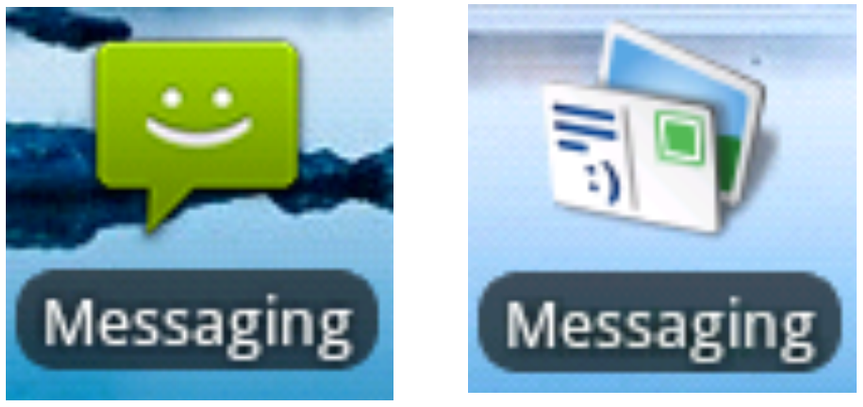


- Work around 1.5 dpi bug
  - Exact matching on 1.5 - 2.0
  - v5 thinks its v6

Resource loading

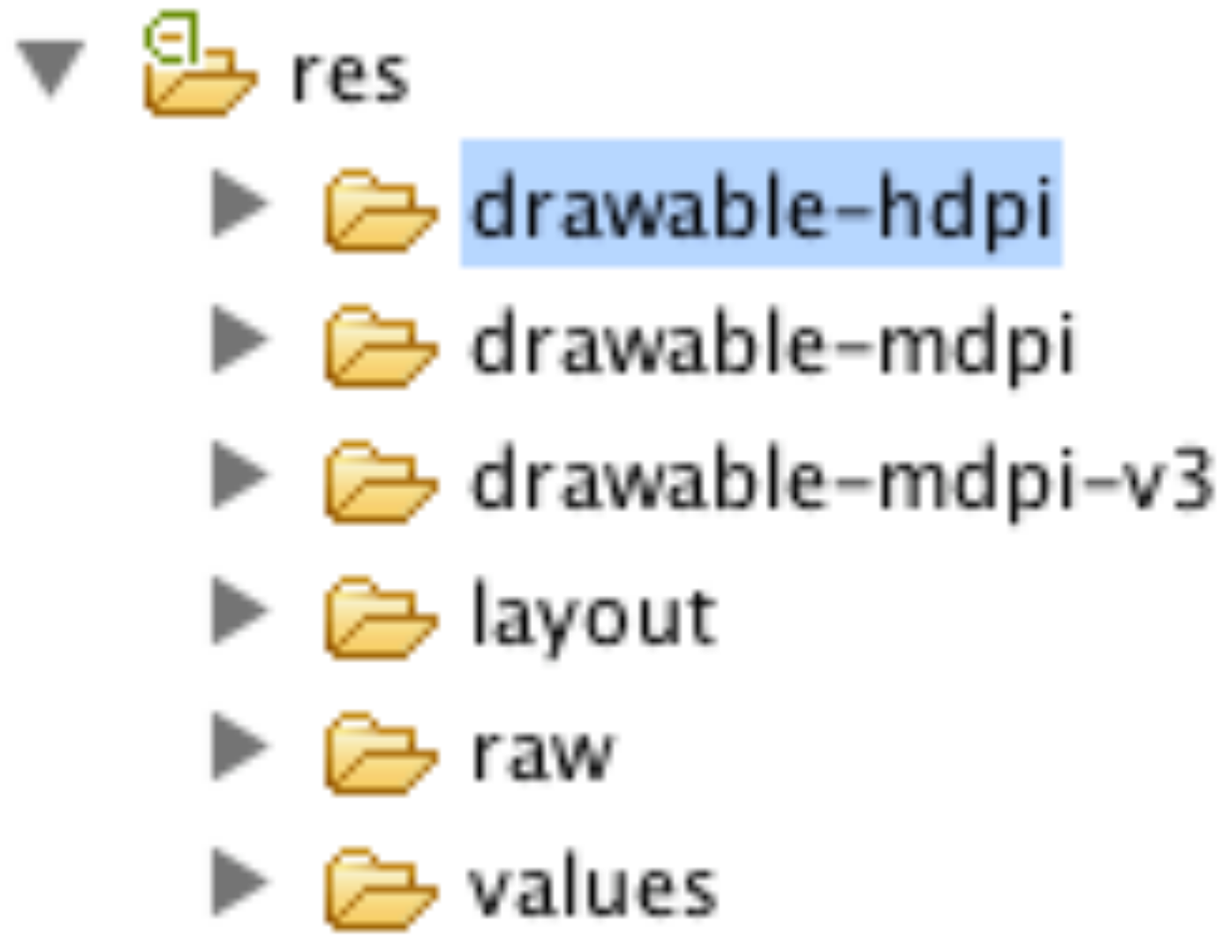
SDK Version

- Styles across versions



- Inexact matching, <= current version

- Work around 1.5 dpi bug
  - Exact matching on 1.5 - 2.0
  - v5 thinks its v6



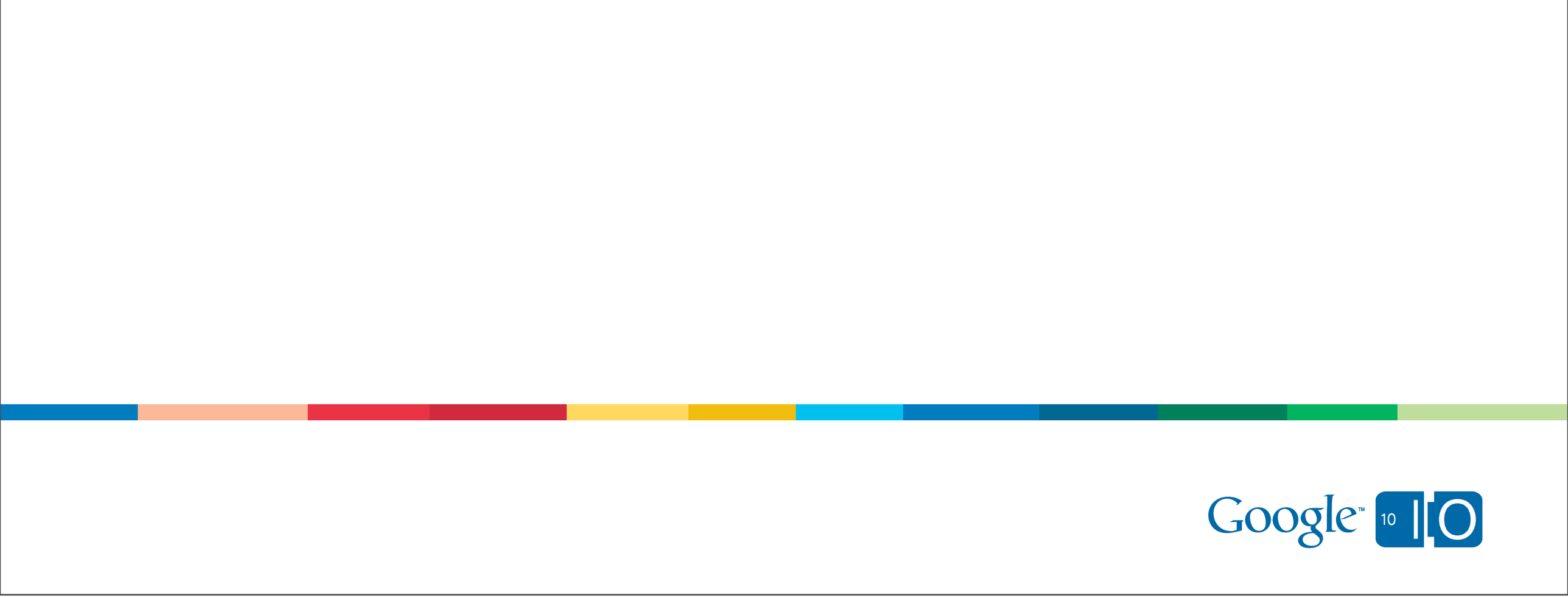

Google™ 10

Image resources

Image Resources

- Goal is uniform *physical* sizing across screens
- Best match allows for up or down sampling based on available resources
  - Single or few pixel features are most obviously impacted

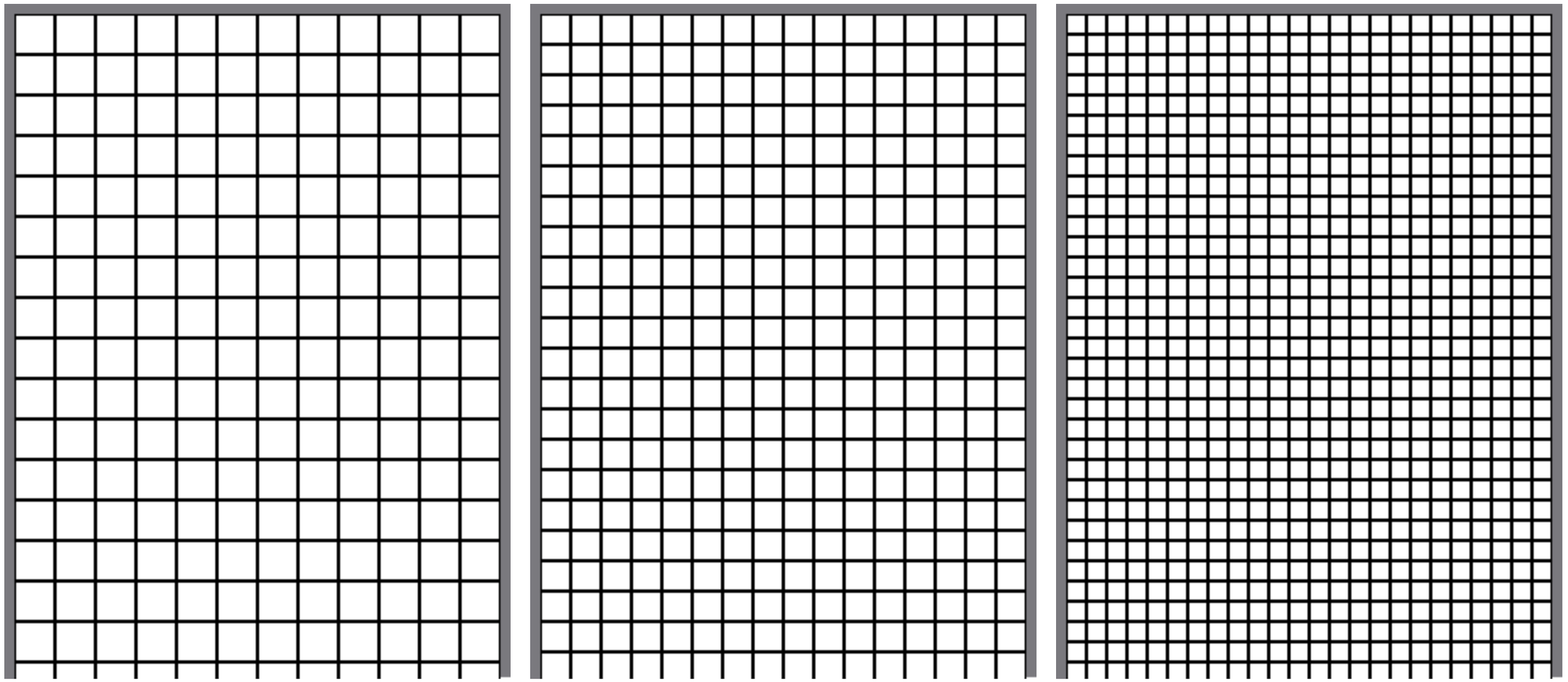


Image Resources

The trouble with pixels

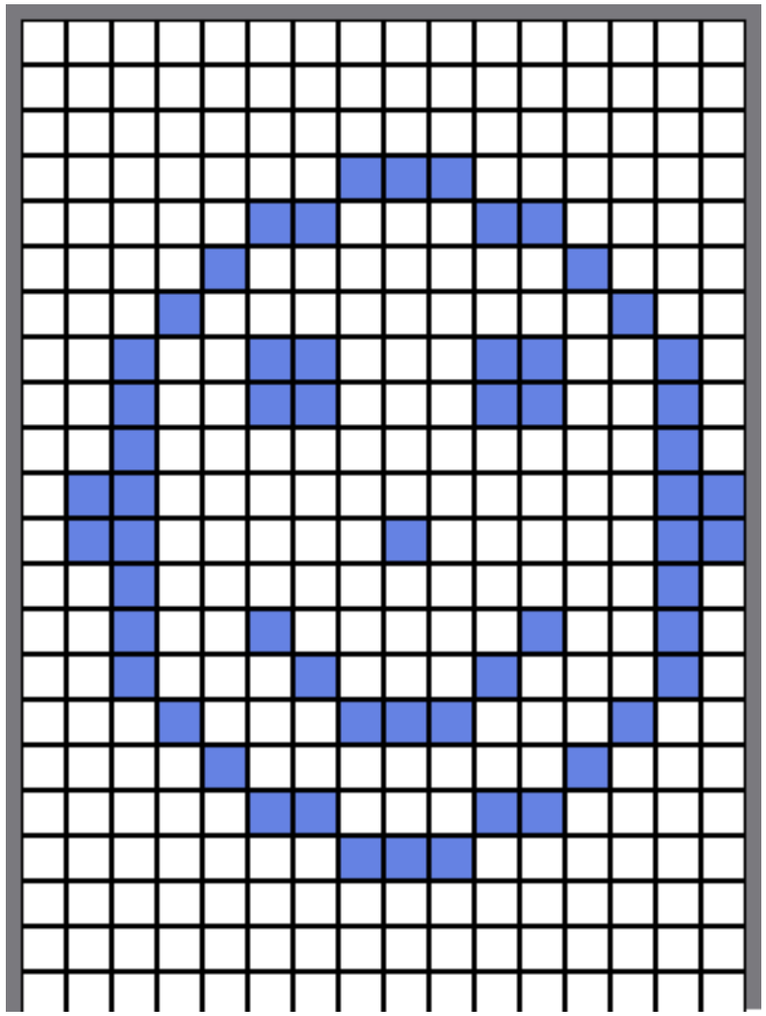
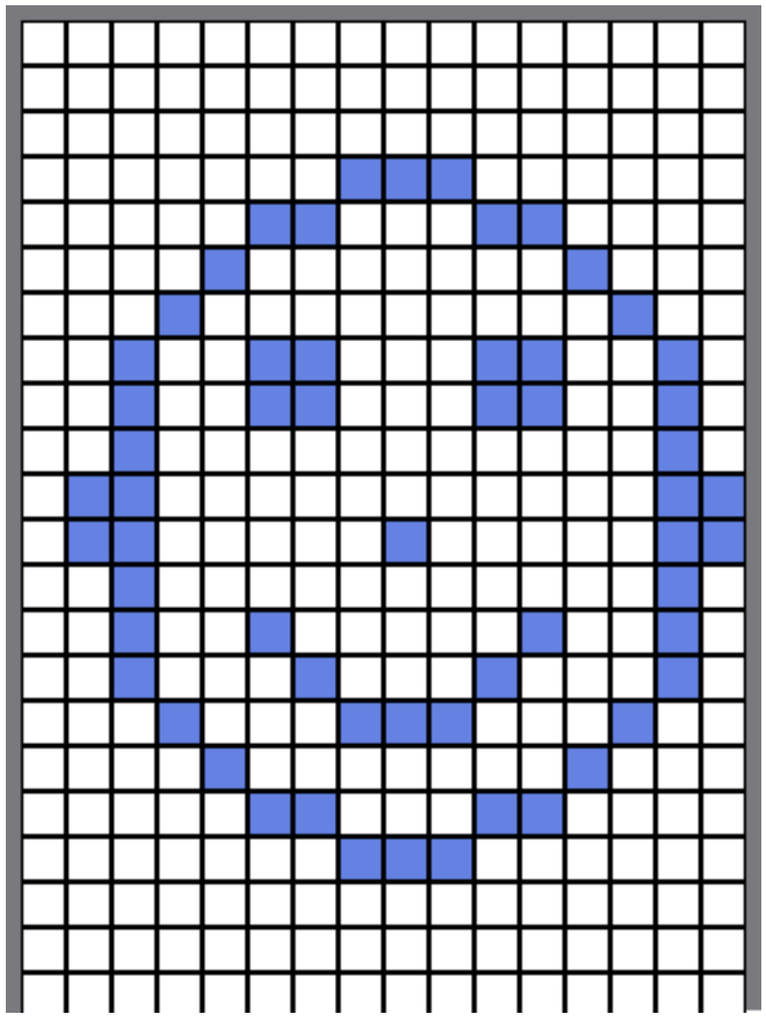


Image Resources

The trouble with pixels



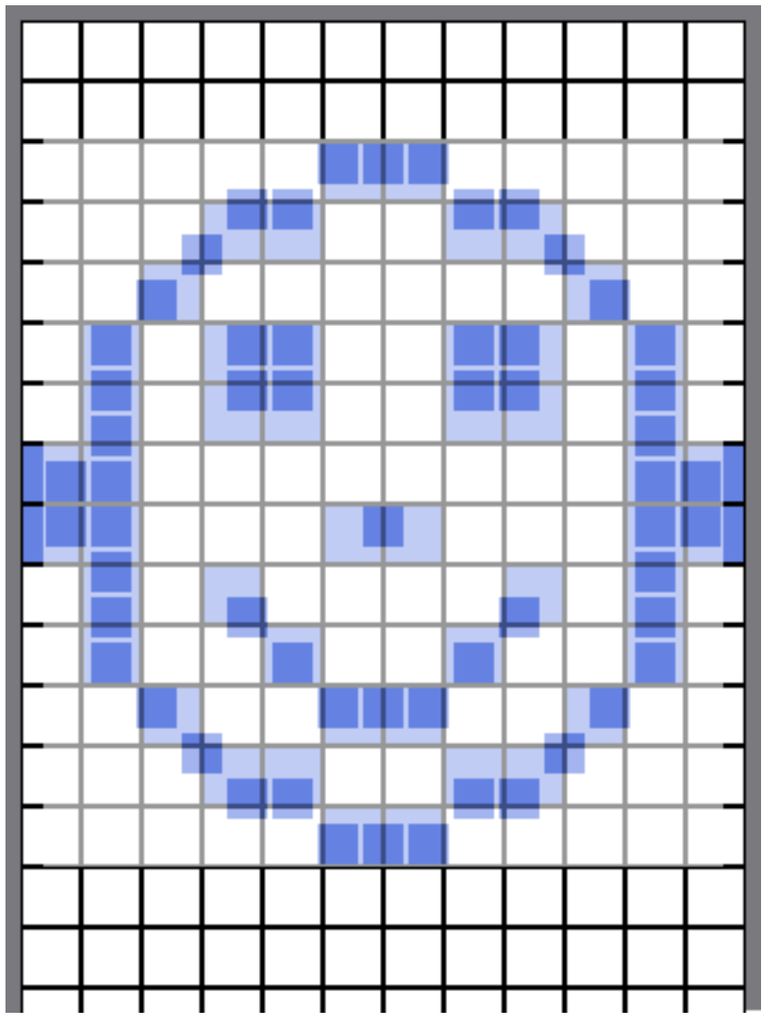
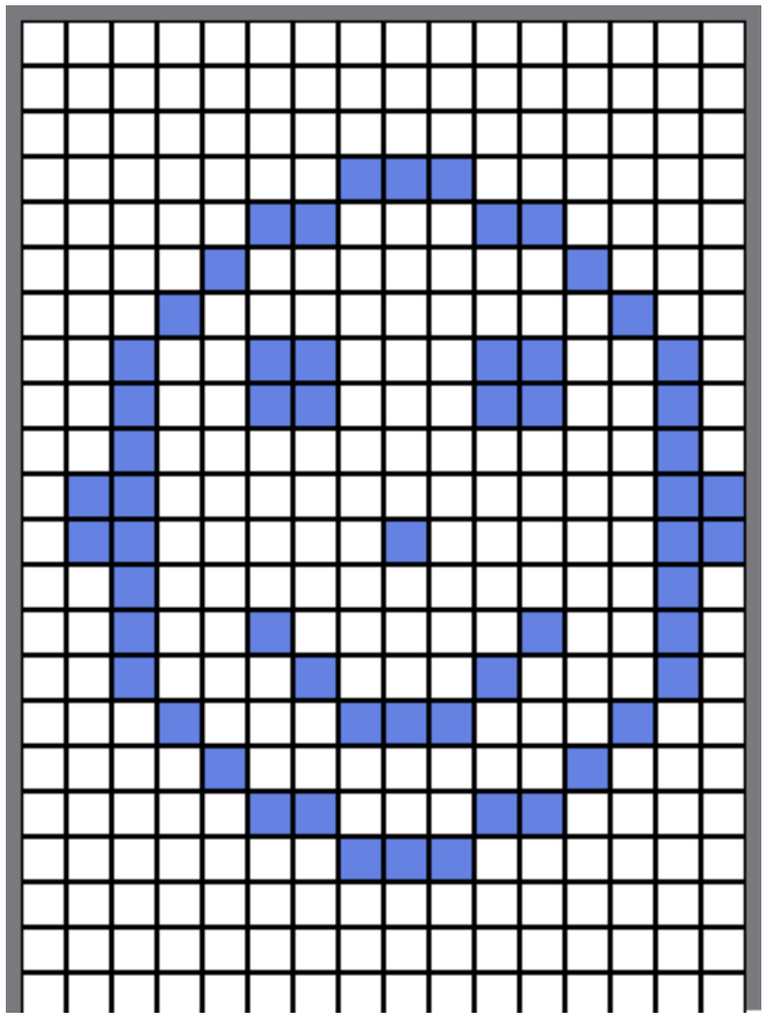


Image Resources

The trouble with pixels



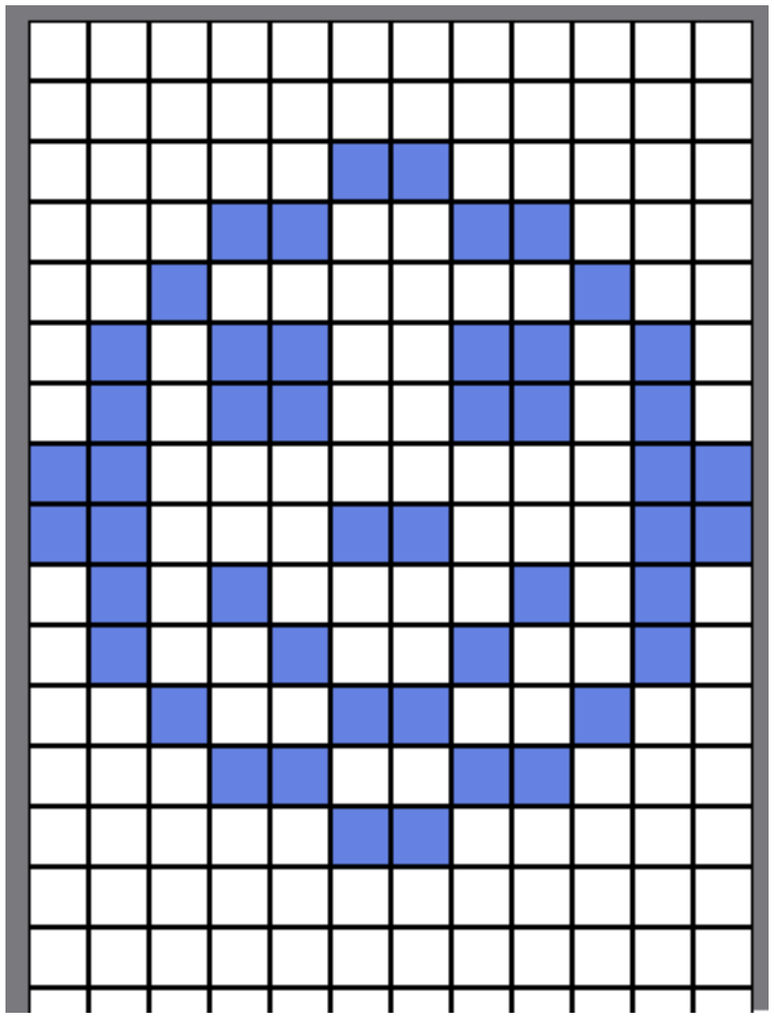
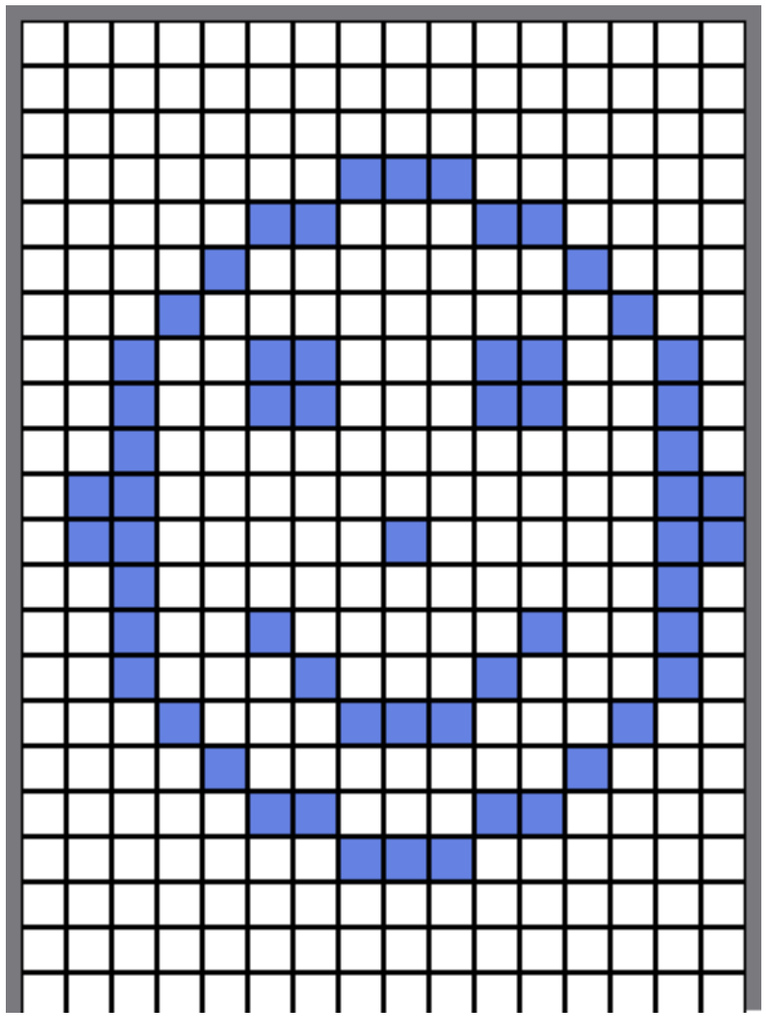
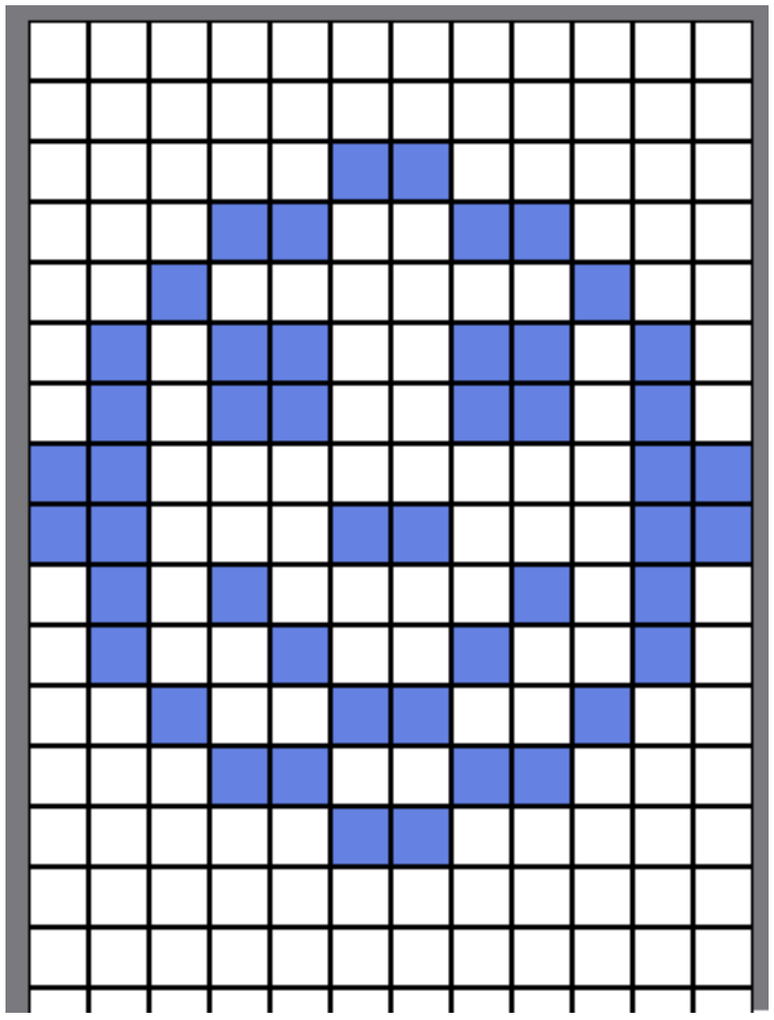


Image Resources

The trouble with pixels





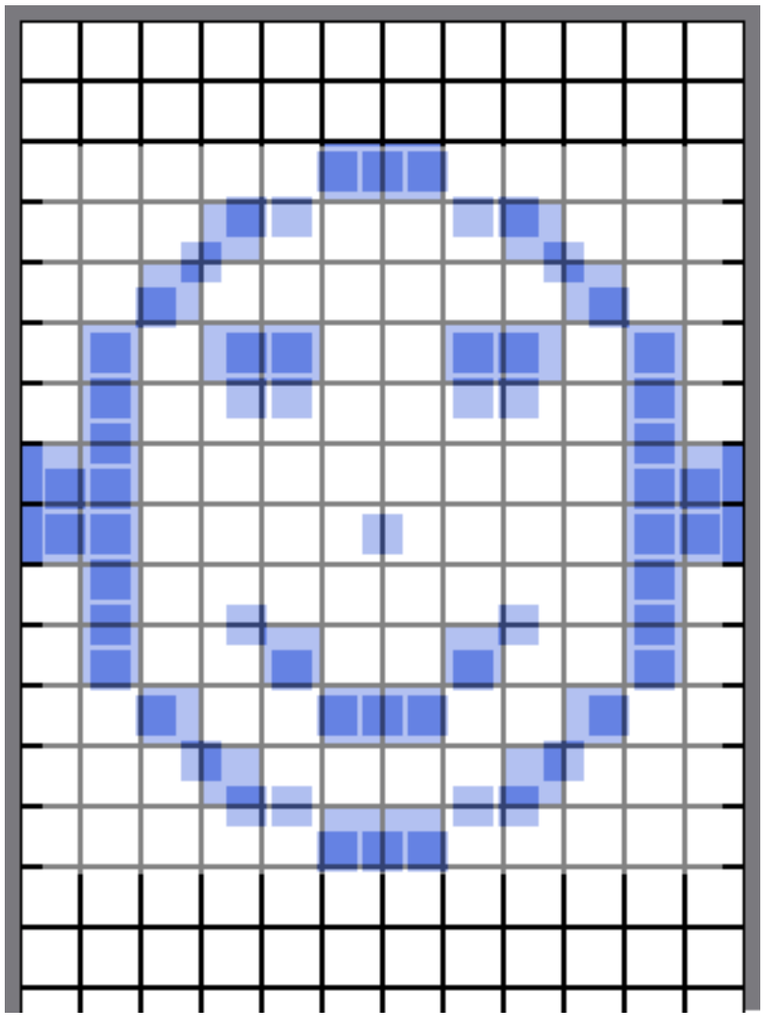
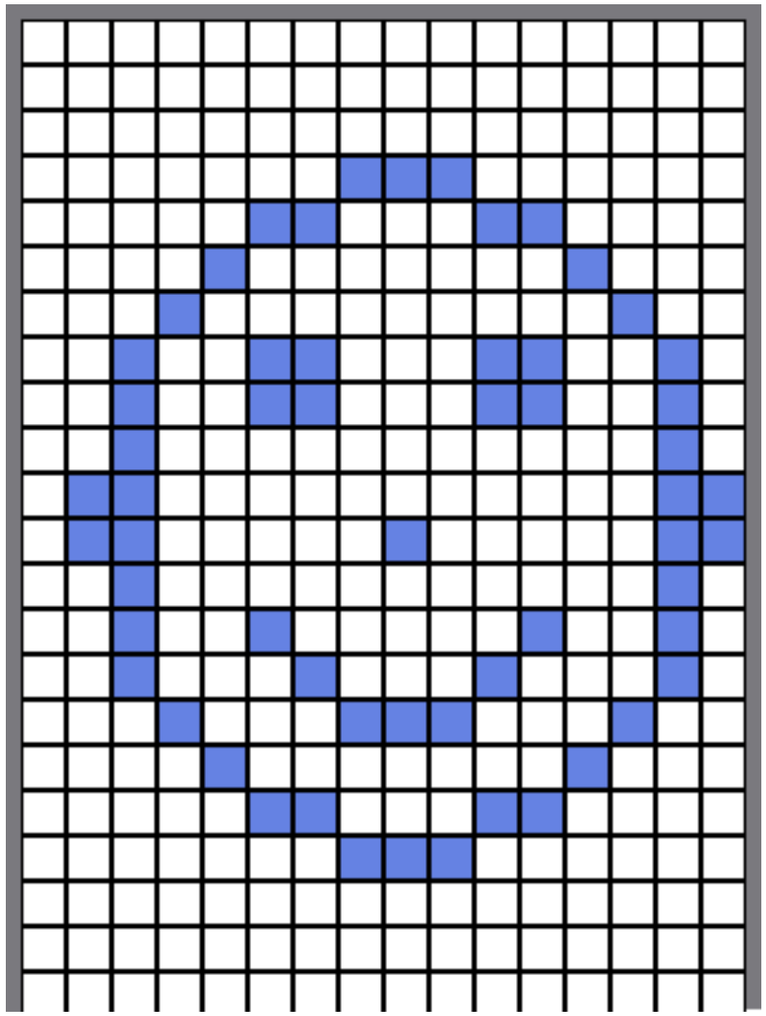
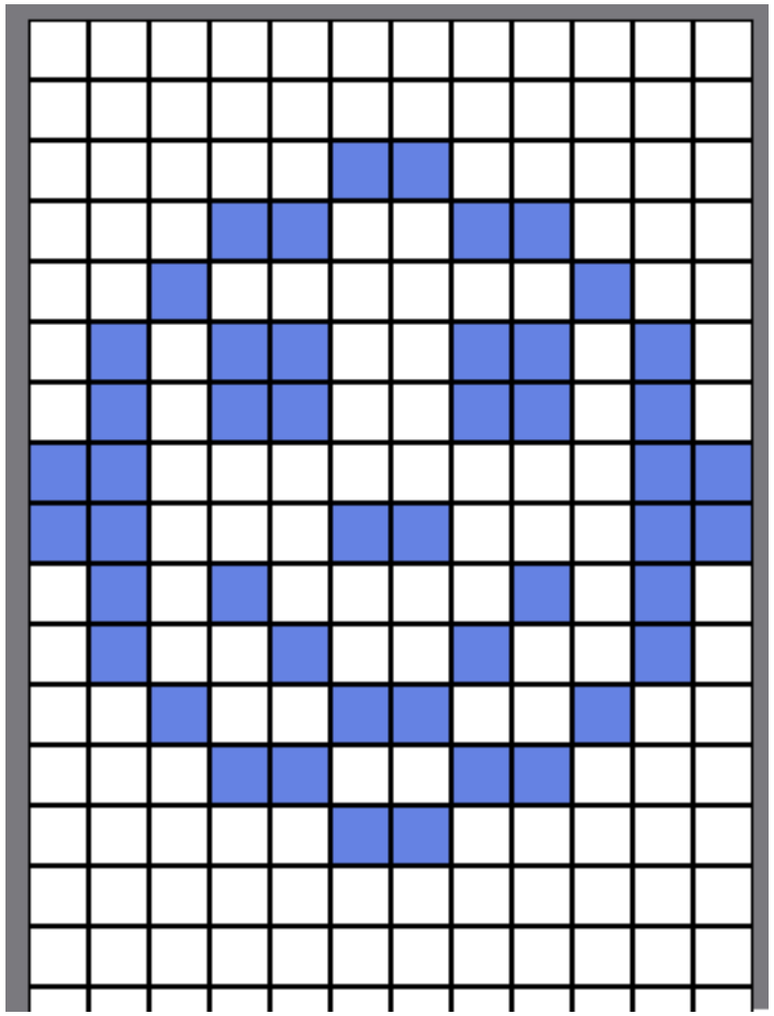
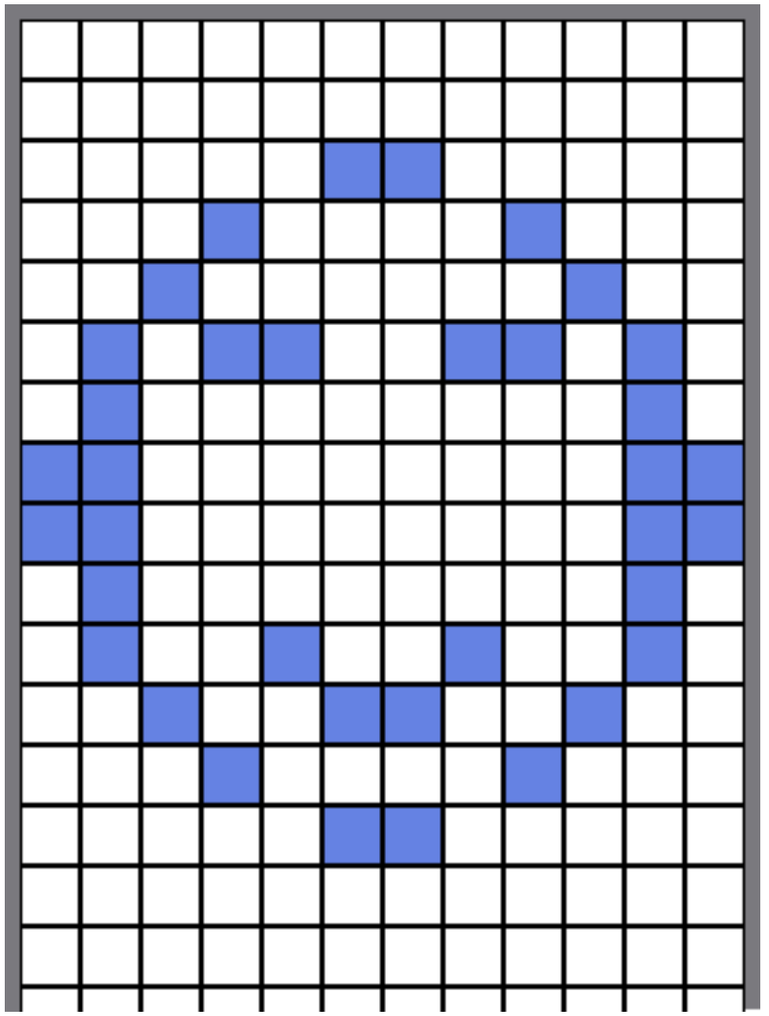


Image Resources

The trouble with pixels







Manipulating Image Loading

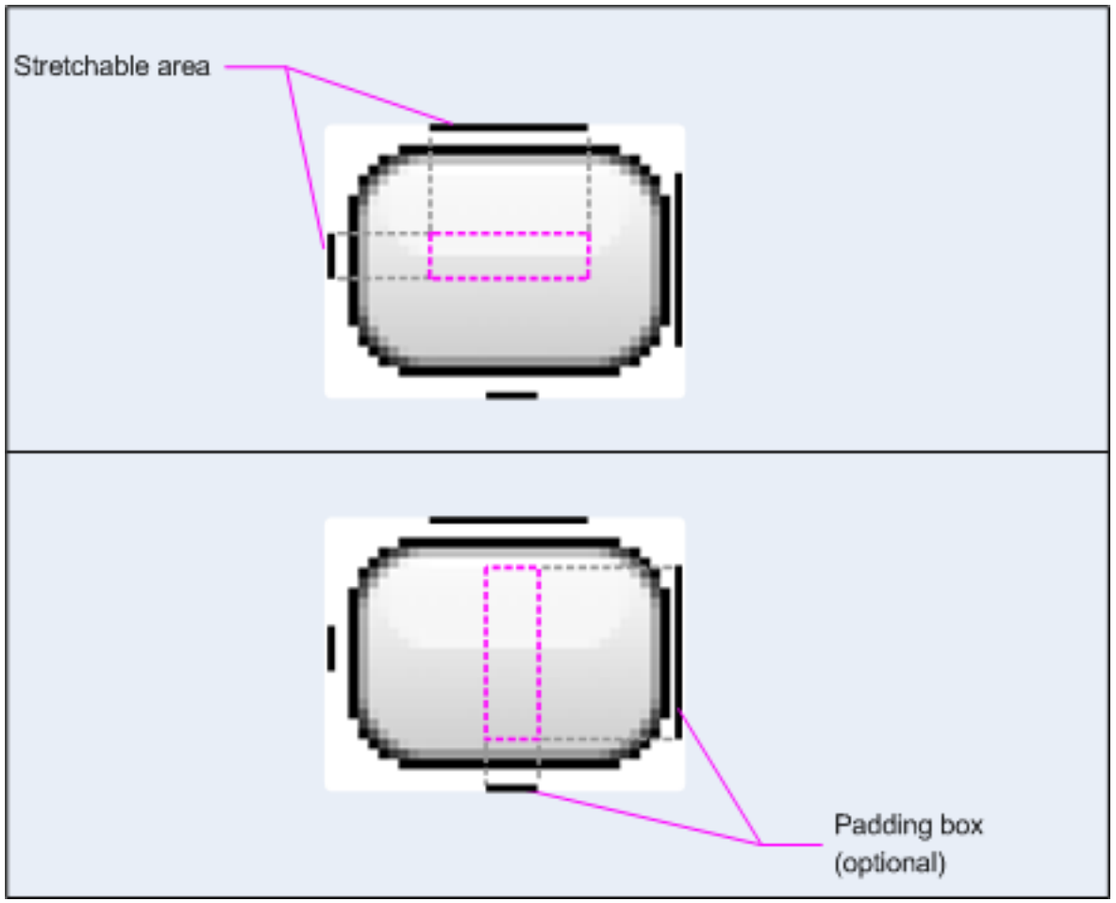
- DO NOT DO THIS
- OKAY, okay, if you must...
- Only possible (and sensical) on 1.6 and above
- Bitmap scaling controlled by density attributes of the object
- Normally...

```
BitmapFactory.Options default = new BitmapFactory.Options();
default.inDensity = <density of loaded resource>
default.inTargetDensity =
getResources().getDisplayMetrics().densityDpi;
```

- Pixels are scaled by inTargetDensity/inDensity
- Manually set inDensity = inTargetDensity = DisplayMetrics.densityDpi

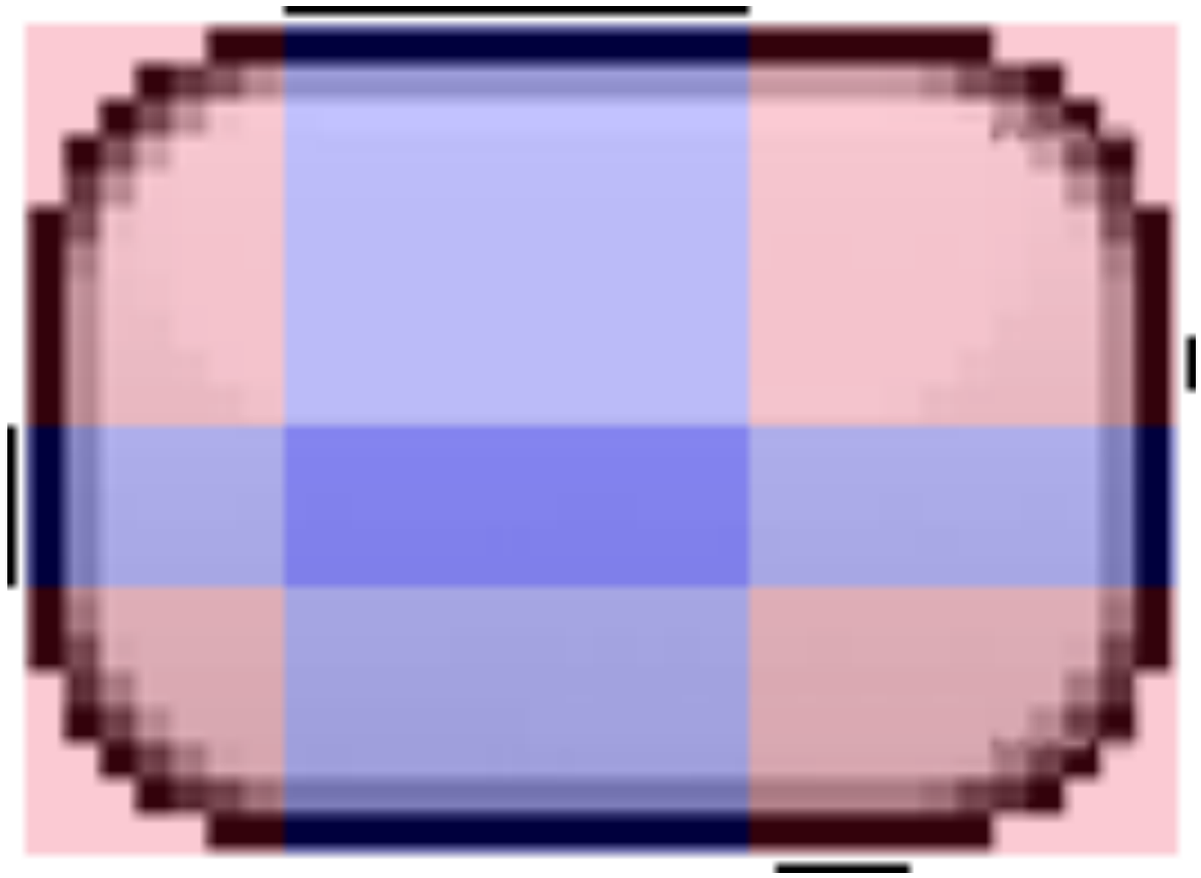
Nine-Patches

- Use to create image resources that can stretch controllably
- One set of controls for stretching, another for the content area



Nine-Patches

Stretching to fill the space

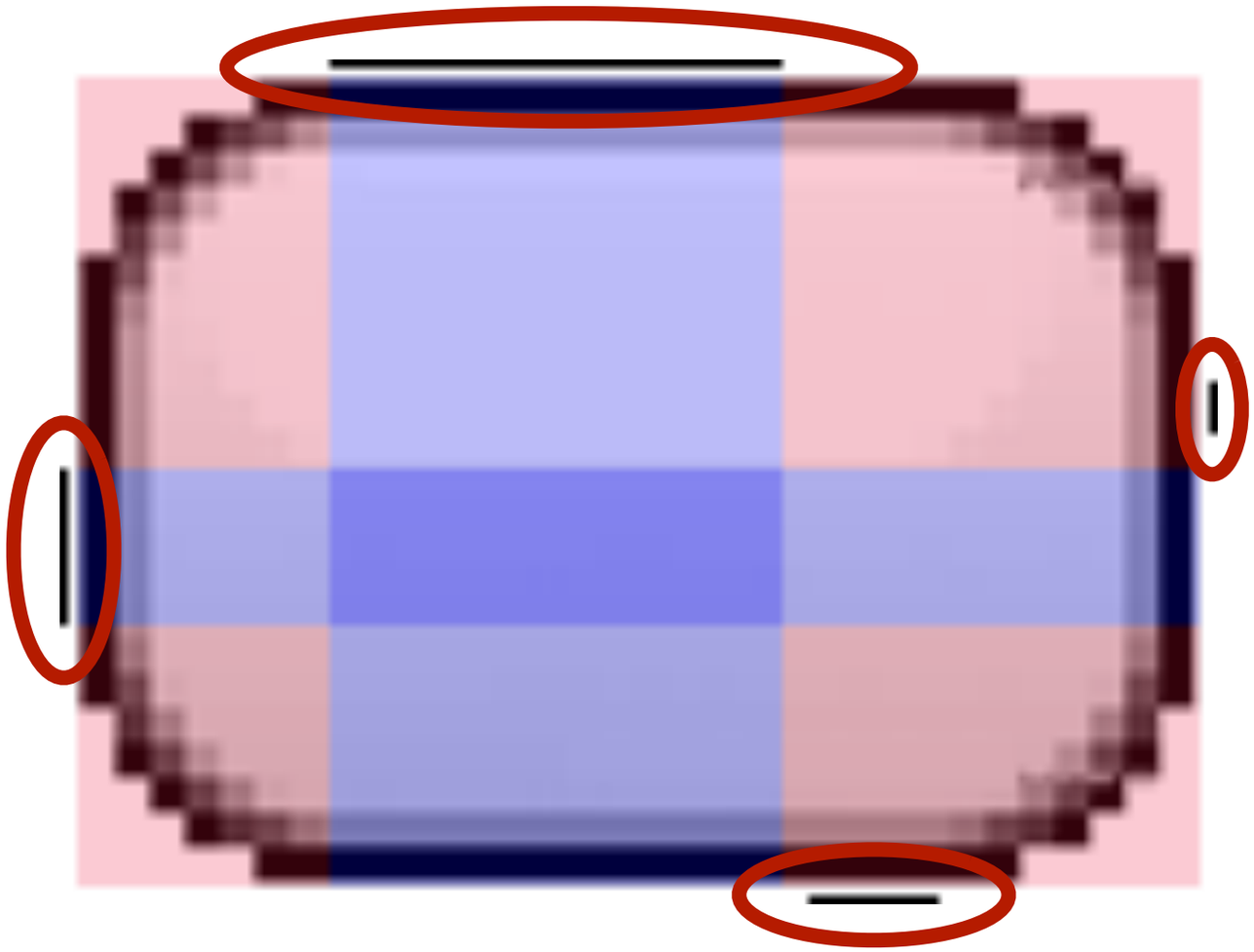


Nine-Patches

Stretching to fill the space

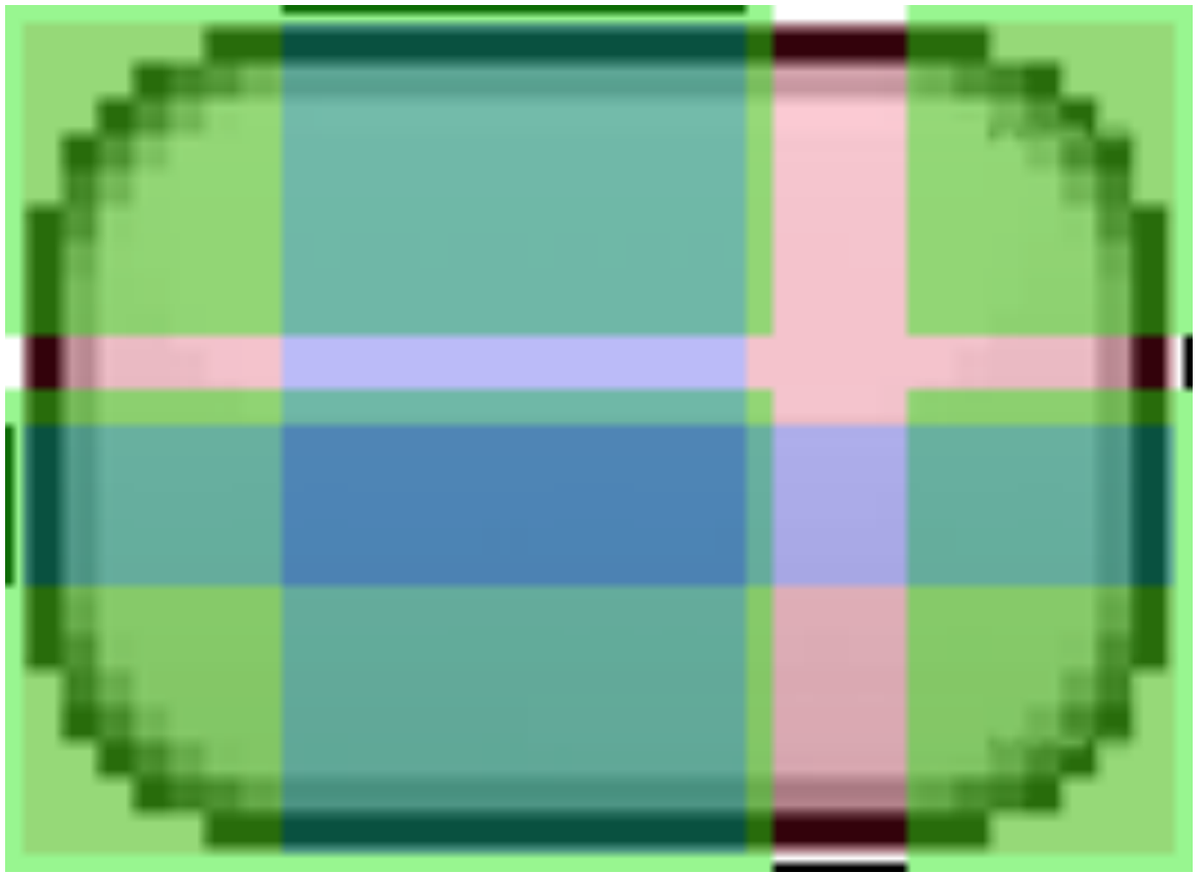
Nine-Patches

Stretching to fill the space



Nine-Patches

Stretching to fill the space



Nine-Patches

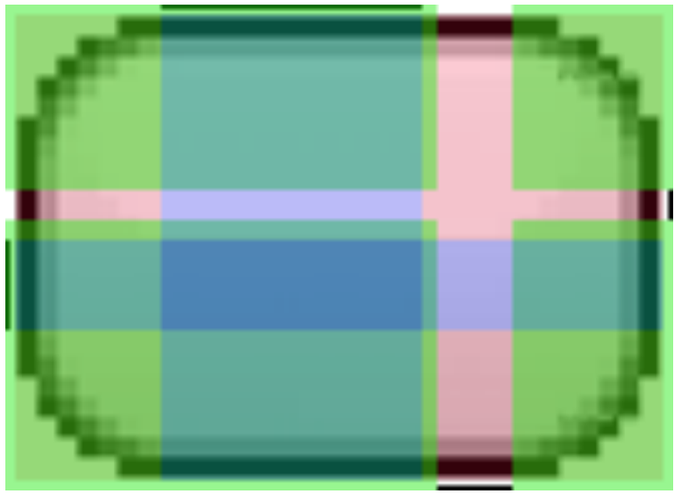
Stretching to fill the space

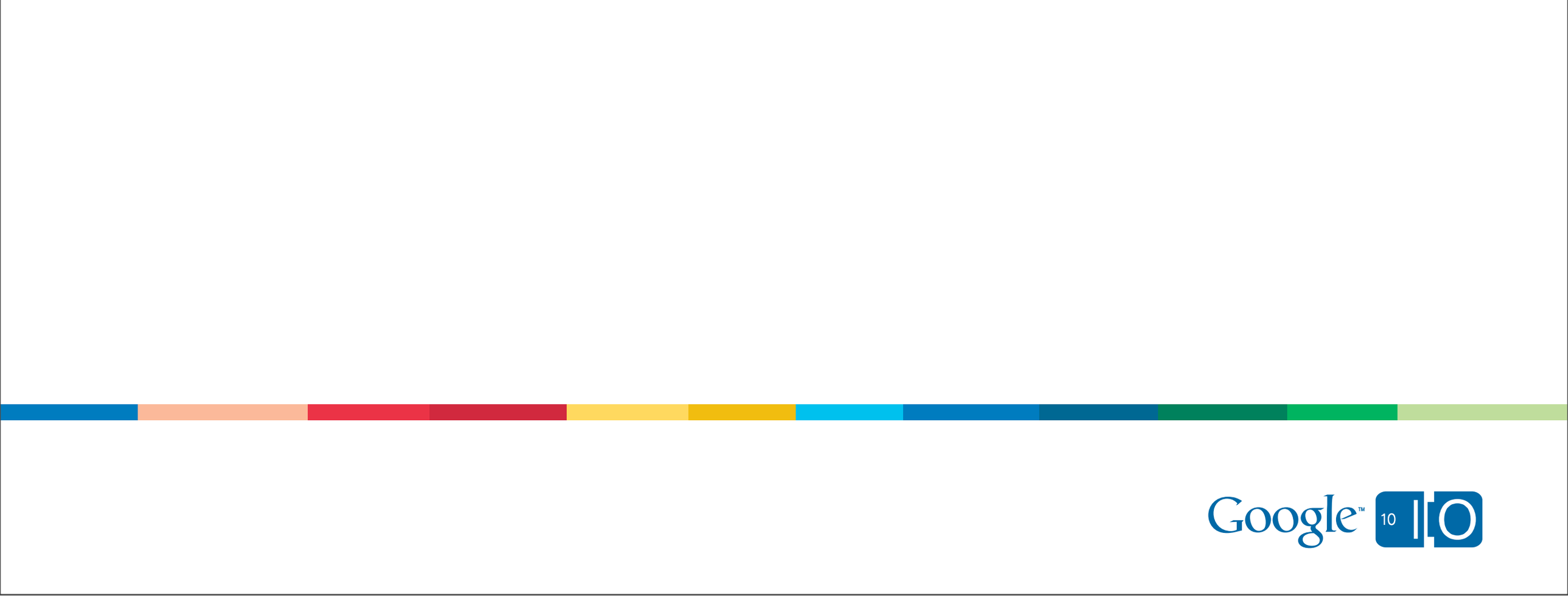
Nine-Patches

Stretching to fill the space

Nine-Patches

A little cushion




Google™ 10

Drawables

Shapes

- Draw optimized views, regardless of screen
- Primitives like square, oval, rectangle
- Use these as the basis for simple graphics instead of bitmaps
- Shape.draw() makes it easy to build a custom view

Shapes

```
public class LogoView extends View {
  private Shape mShape = new RectShape();
  private Drawable mLogo;
  public LogoView (Context context) {
    super(context);
    mLogo = getContext().getResources()
        .getDrawable(R.id.logo);
  }
  protected void onDraw(Canvas canvas) {
    mShape.resize(canvas.getWidth(), canvas.getHeight());
    mShape.draw(canvas, mLogoBackgroundPaint);
    mLogo.draw(canvas);
  }
}
```

Google™ 10 I/O

ScaleDrawable

- Adapt your drawables in a custom manner based on screen
- Wraps another drawable
- Allows for more exact control over scaling
- Ideal for situations where it doesn't make sense to preserve aspect ratio
  - Progress bars
  - Dividers
  - Any place where one dimension is more important than the other

LevelListDrawable

- Specified in XML
- Combines a group of resources into a drawable
- Use where the drawable is logically the same, but presentation varies based on state
  - status meters (eg. battery, signal)
  - weather condition
  - sports team
  - UI theme
- For cases without a natural “level”, define constants making code more readable
- Select the right resource for the right UI theme

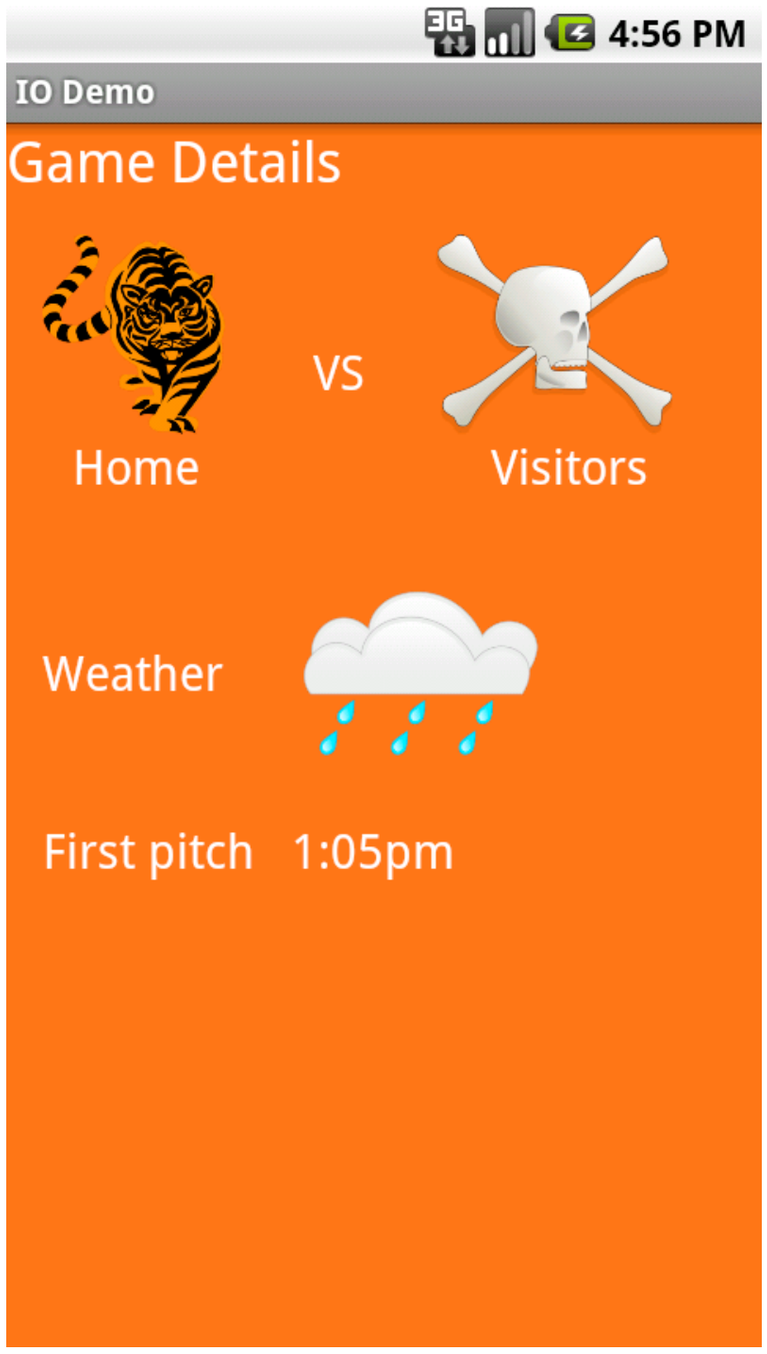
LevelListDrawable

```
<?xml version="1.0" encoding="utf-8"?>
<level-list xmlns:android="http://schemas.android.com/apk/
res/android">
  <item android:maxLevel="0"
      android:drawable="@drawable/rain_chance_zero"/>
  <item android:maxLevel="10"
      android:drawable="@drawable/rain_chance_slight"/>
  <item android:maxLevel="40"
      android:drawable="@drawable/rain_chance_good"/>
  <item android:maxLevel="70"
      android:drawable="@drawable/rain_chance_high"/>
</level-list>
```

LevelListDrawable

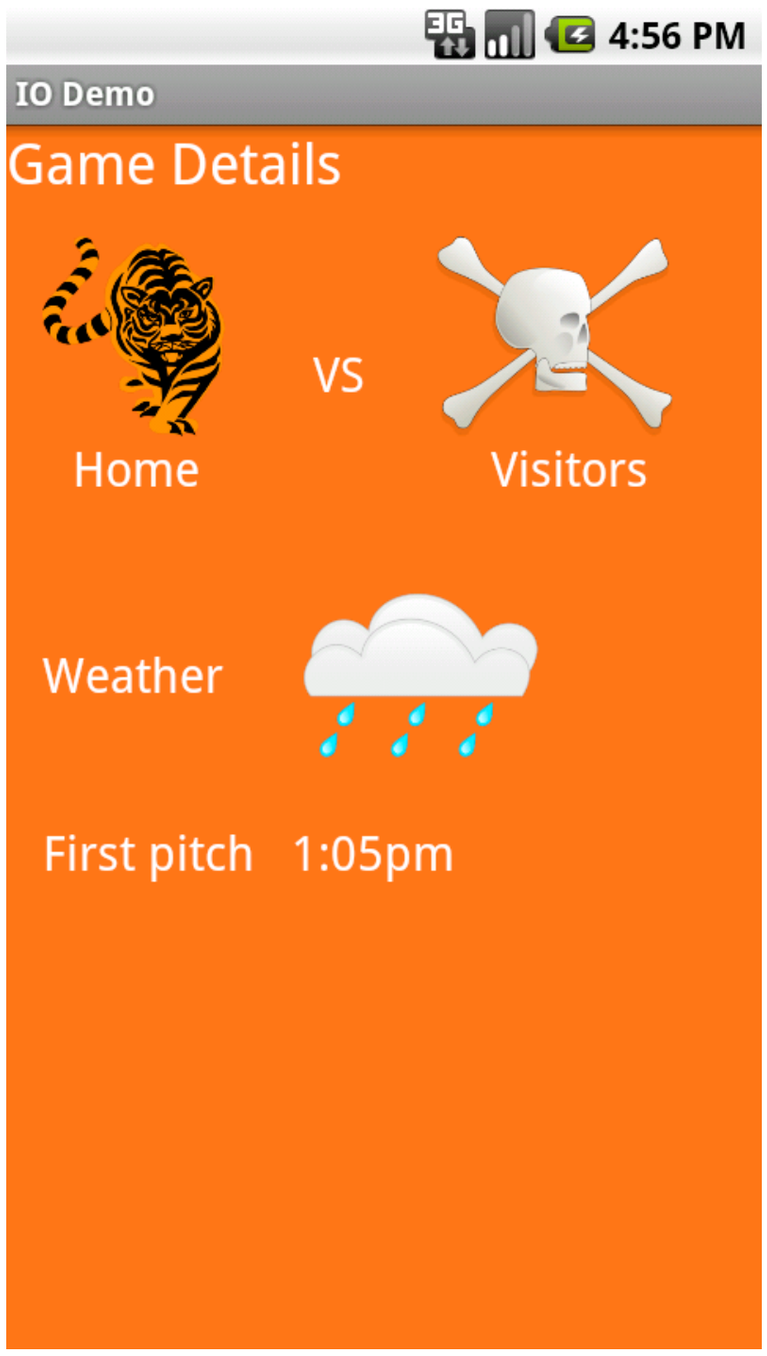
```
public class IO_Demo extends Activity {
  public void onResume() {
    ((ImageView)findViewBy(R.id.weather))
        .setImageLevel(getRainChance());
  }

  private int getRainChance() {
    // call web service to get forecast
  }
}
```

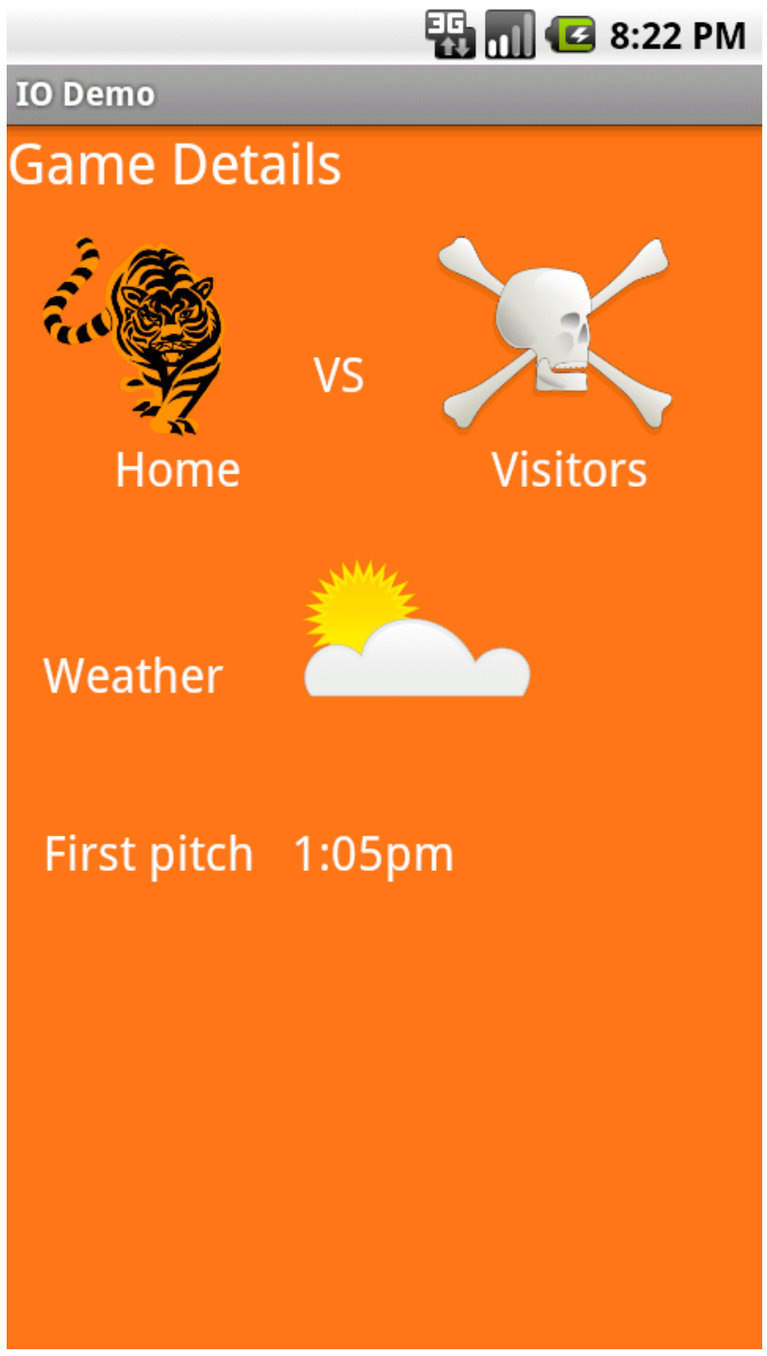
LevelListDrawable



Thanks to openclipart.org from for the Public Domain clipart used on this slide.

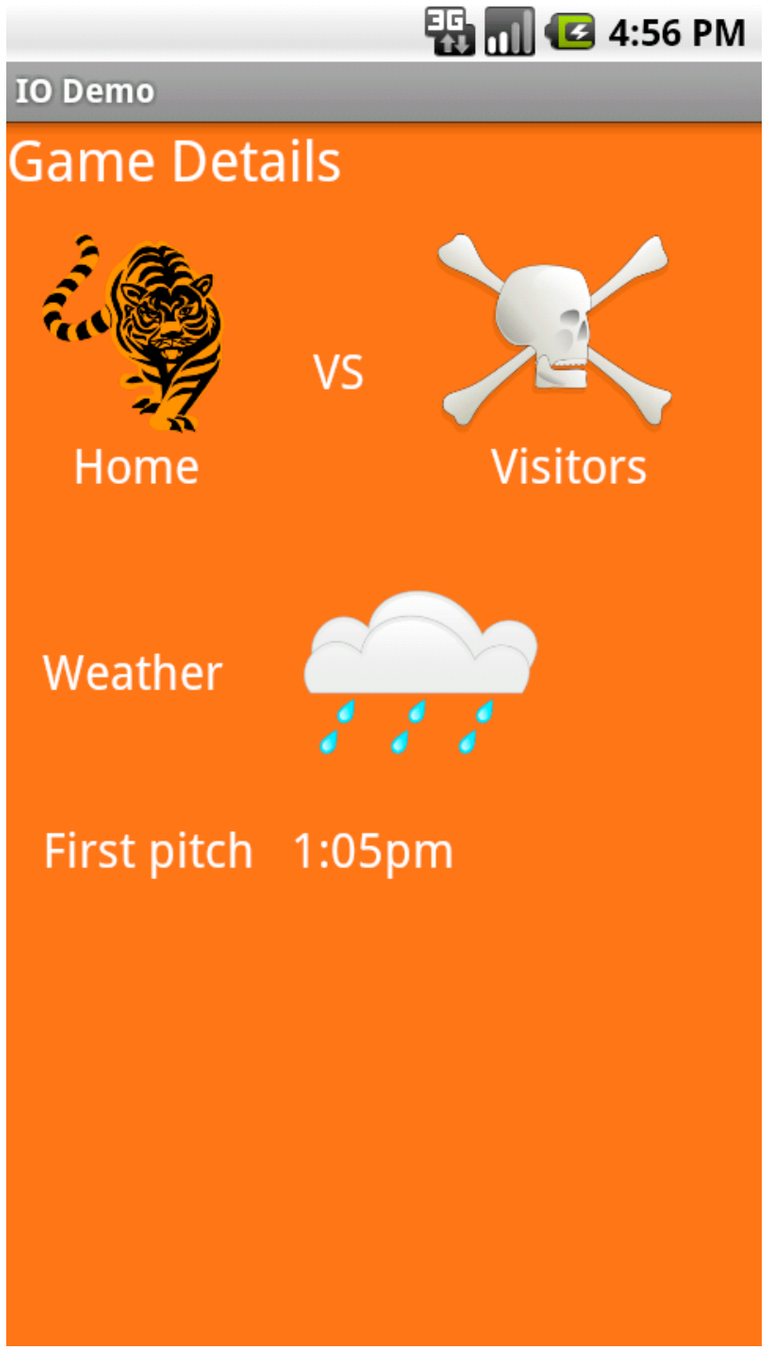
LevelListDrawable

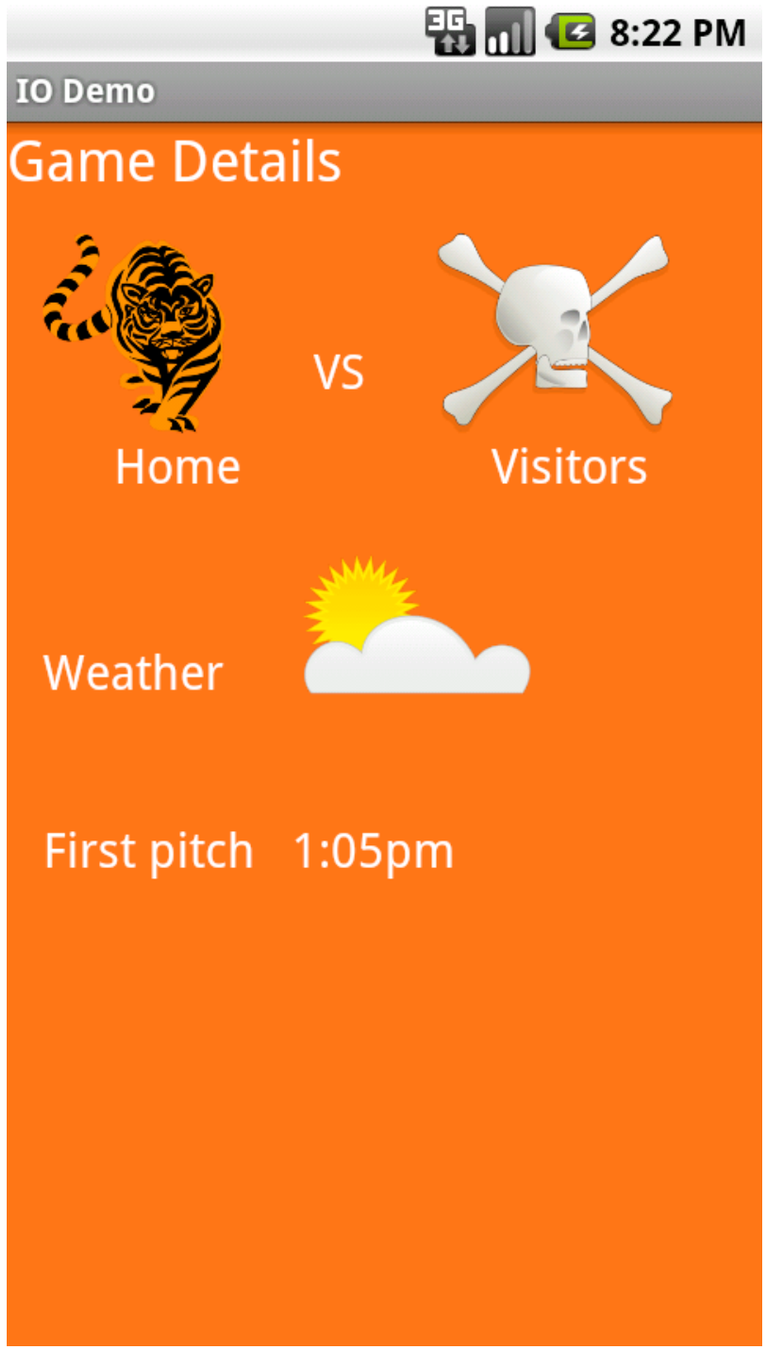


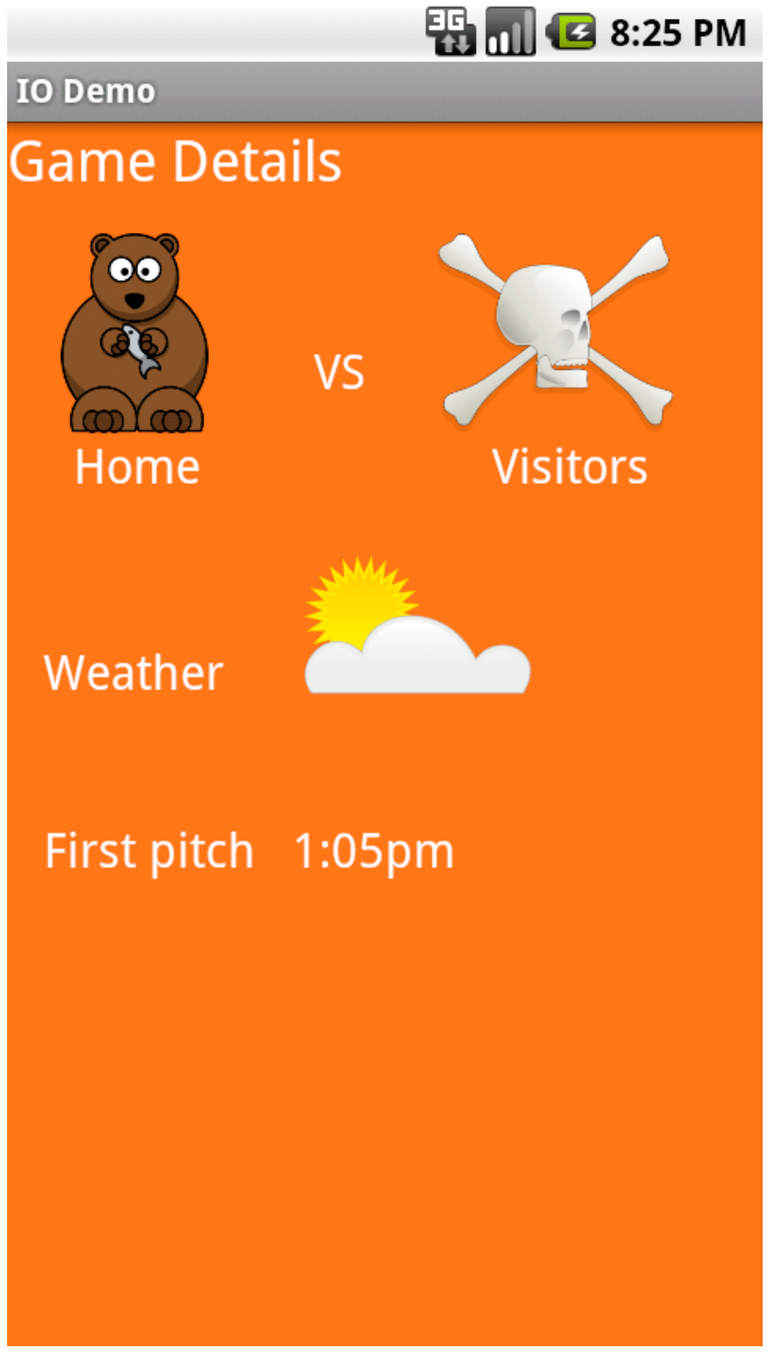


Thanks to openclipart.org from for the Public Domain clipart used on this slide.

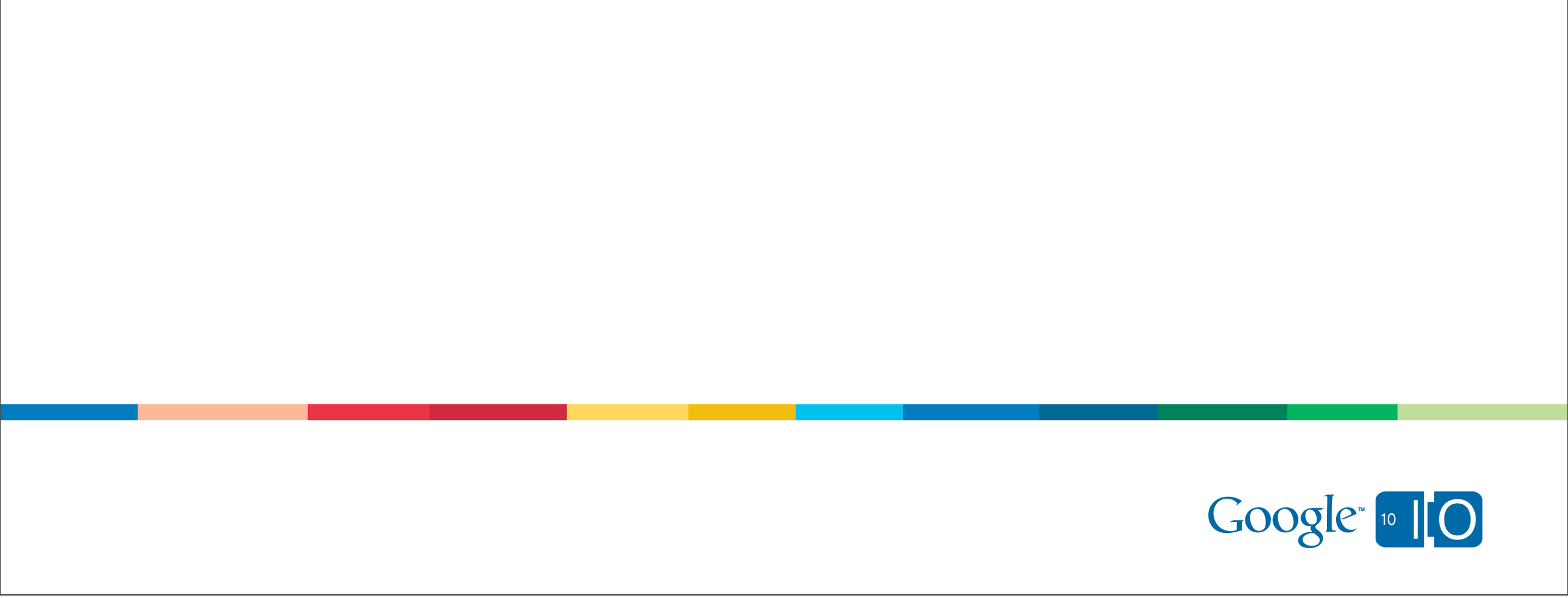
LevelListDrawable







Thanks to openclipart.org from for the Public Domain clipart used on this slide.


Google™ 10

Cross-Version Compatibility

API Changes: New Classes

- Runtime handling of class load failures
- Class dependencies resolved when class is first loaded
- Use a second class with a static method to provoke class verification
- Catch any verification error and adapt behavior
- Avoids messy Class.forName(“”) pattern, but grows number of classes needed
  - group in a single package or as inner classes

API Changes: New Classes

```
public class Canary {
  private static NEW_CLASS foo;

  public static void tryNewClass() {
    foo = new NEW_CLASS();
  }
}
```

API Changes: New Classes

```
public class MyActivity extends Activity {
  private boolean canaryAvailable;
  private void checkApis() {
    try {
      Canary.tryNewClass();
      canaryAvailable = true;
    } catch (VerifyError e) {
      canaryAvailable = false;
      Log.w(LOG_TAG, “Canary unavailable, falling back.”);
    }
  }
```

API Changes: New Methods

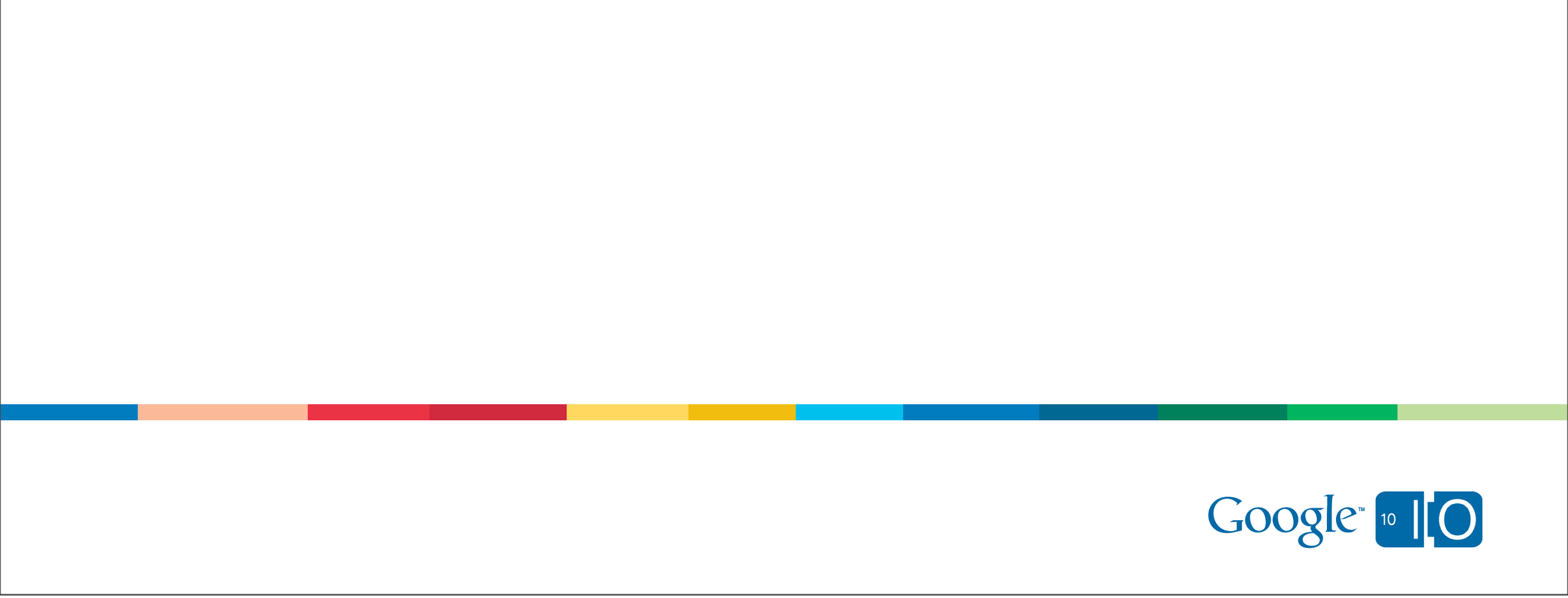
- A bit trickier than dealing with classes
- Determining which APIs are available
  - Use reflection to check availability
  - Make your code version aware
- Working with availability
  - Call through using reflection (fewer classes)
  - Write a factory to return the right class (cleaner)

API Changes: Mirror, mirror...

```
public class Canary {
  private static Method getDensity;
  static {
    getDensity = Canvas.class.getMethod(“getDensity”, null);
  } catch (NoSuchMethodException e) {
    Log.w(LOG_TAG, “getDensity method not available.”);
  }
}
```

API Changes: Mirror, mirror...

```
public class Canary {
  private static boolean hasDensity;
  private Canvas mDrawingSurface;
  public Integer getDensity() {
    if (getDensity != null) {
      return (Integer) getDensity.invoke(mDrawingSurface,
          null);
    } else {
      return DEFAULT;
    }
  }
```


Google™ 10

Testing

Google I/O 10

Testing, testing, testing

- Nothing beats it
- Verify the UI is behaving as expected
- Check that any API compatibility code functions properly
- Hardware testing is hopefully unnecessary, but...

Questions?

Use Wave to see live notes and ask questions

http://tinyurl.com/io10casting