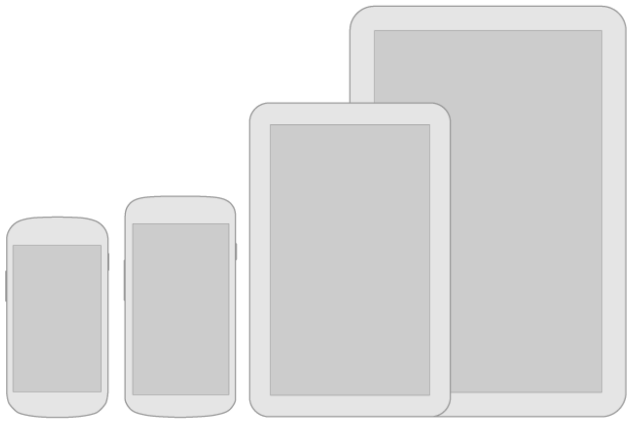
SunLi@MrSunLi.com
http://MrSunLi.com

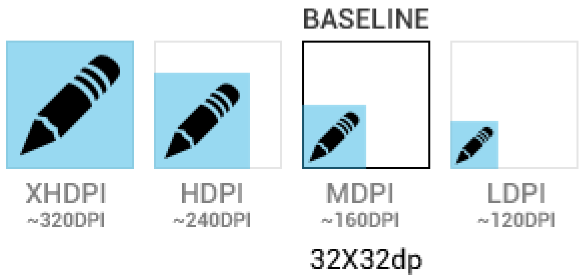
# Supporting Multiple Screens On Android (covers 4.0)

## Target Audiences

- Designer for graphics assets for Android devices
- Designer for graphics assets for Android apps
- Beginning developers for Android apps

## Agenda

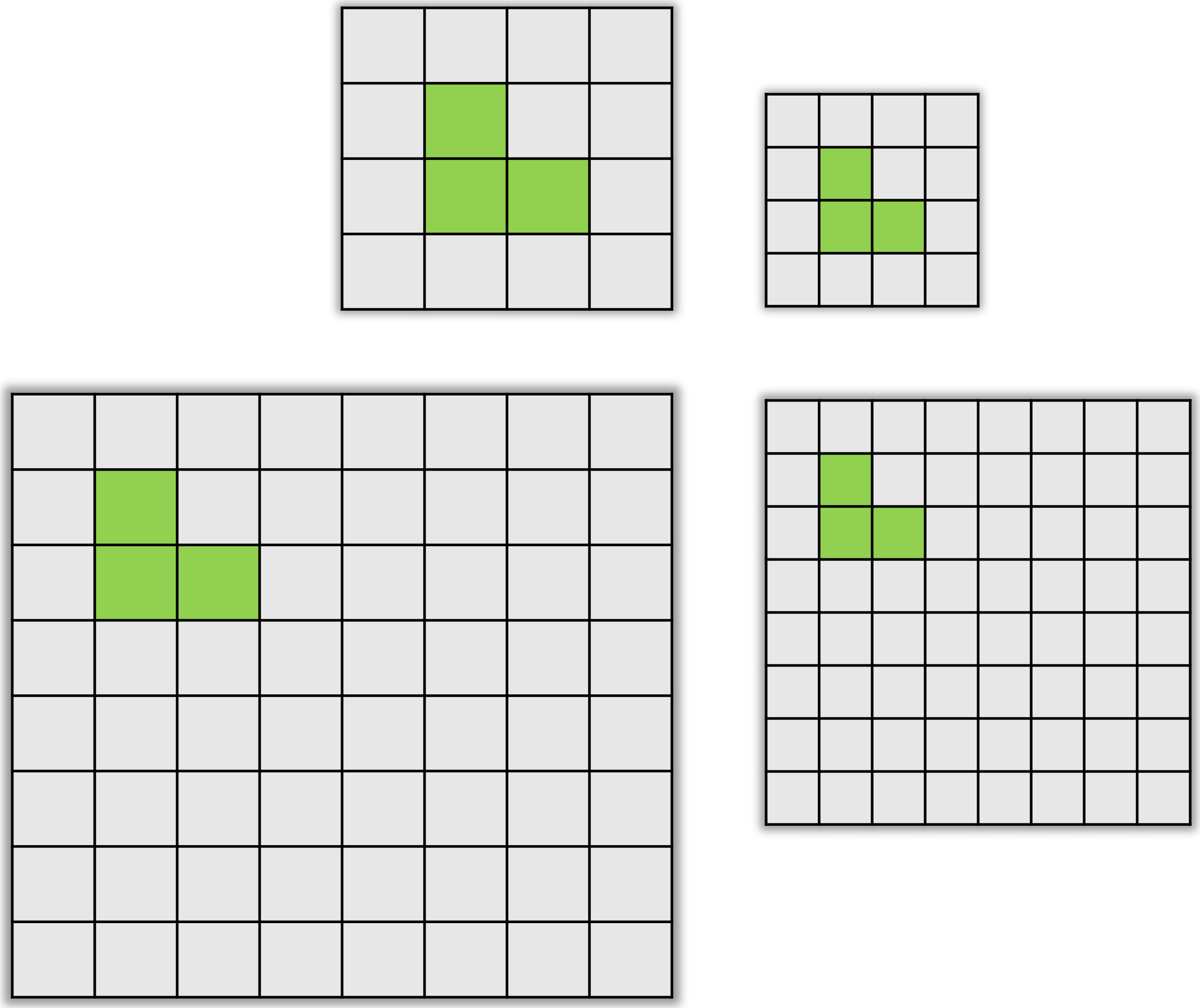




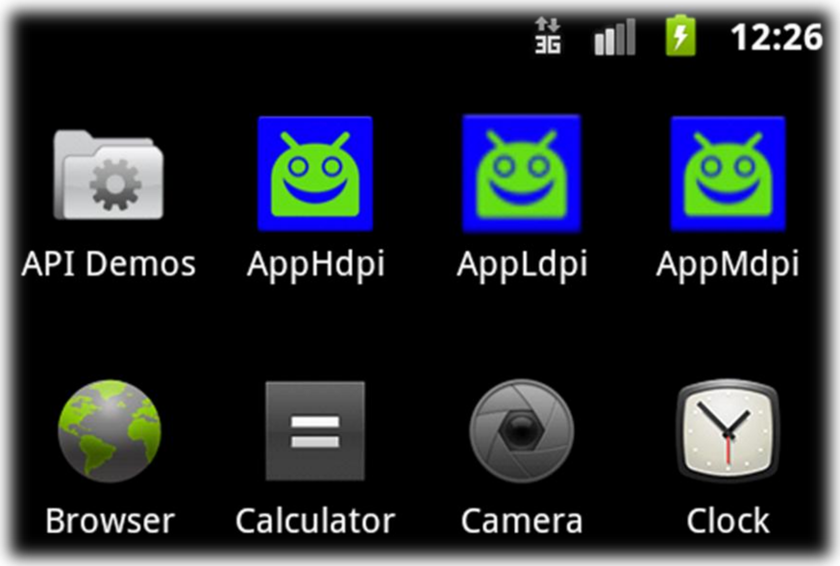
- Challenges
- Solutions
- Web resources
- **Key takeaway**

- Technical details
  - Technical terms and concepts
  - Range of screens supported
  - Alternative graphics assets for different screen densities
  - Web resources

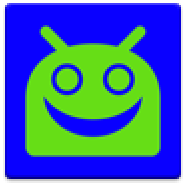
# Resolution independence and density independence are critical challenges



## Resolution is well-managed by Android, but density might blur graphics assets



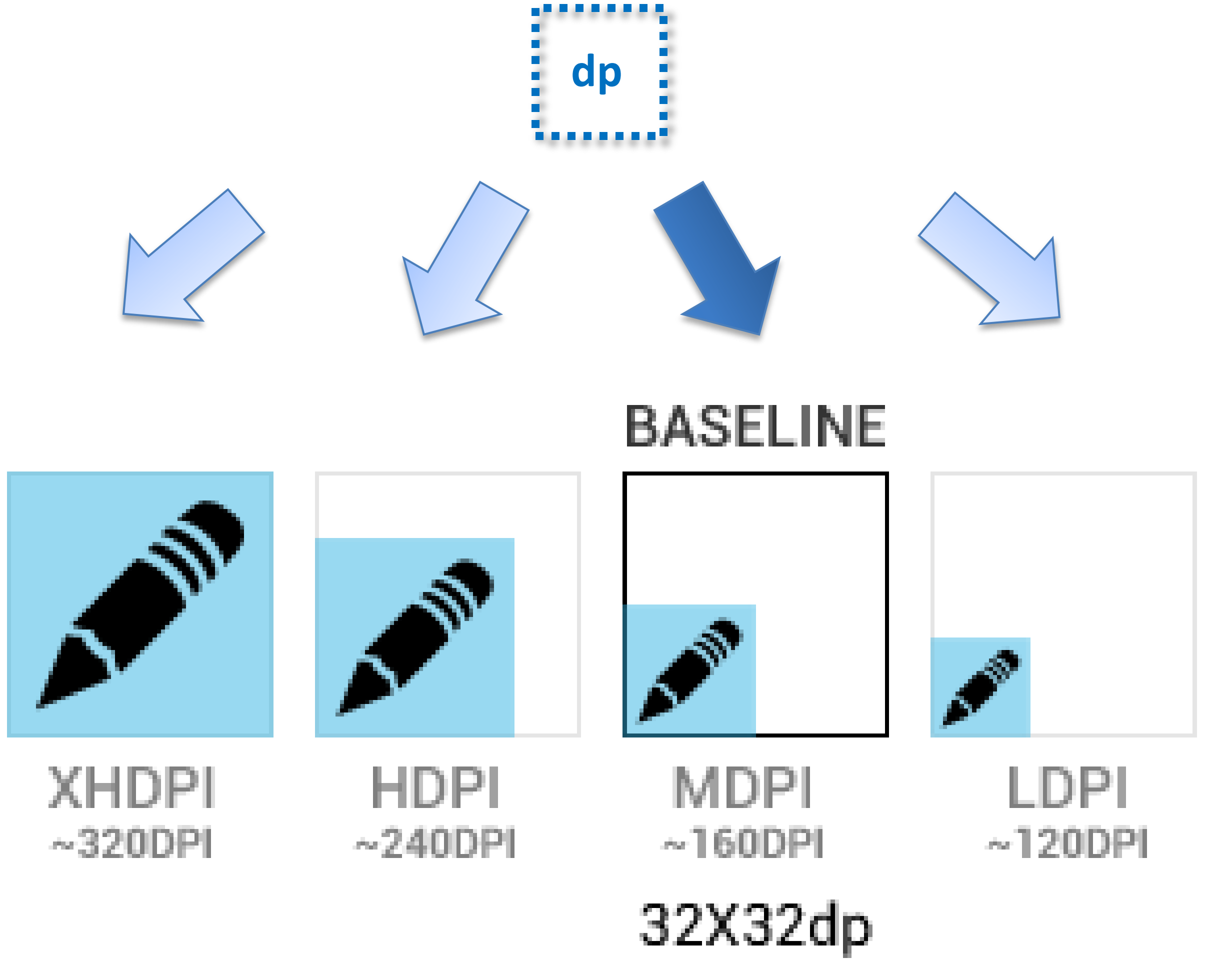
Screen snapshot on a **hdpi** device

- **ldpi** = 120
  - 36 x 36 **px**
- **mdpi** = 160
  - 48 x 48 **px**
- **hdpi** = 240
  - 72 x 72 **px**
- **xhdpi** = 320 
  - 96 x 96 **px**

## Density can be managed by Android with standard categories, although device display panels are diversified

| | | |
|---|---|---|
| | Wildfire | **125 dpi** |
| | Xoom | **149 dpi** |
| | Nexus S | **235 dpi** |
| | Galaxy Nexus | **316 dpi** |

## Density-independent pixel (dp) is a new virtual unit for conceptual design

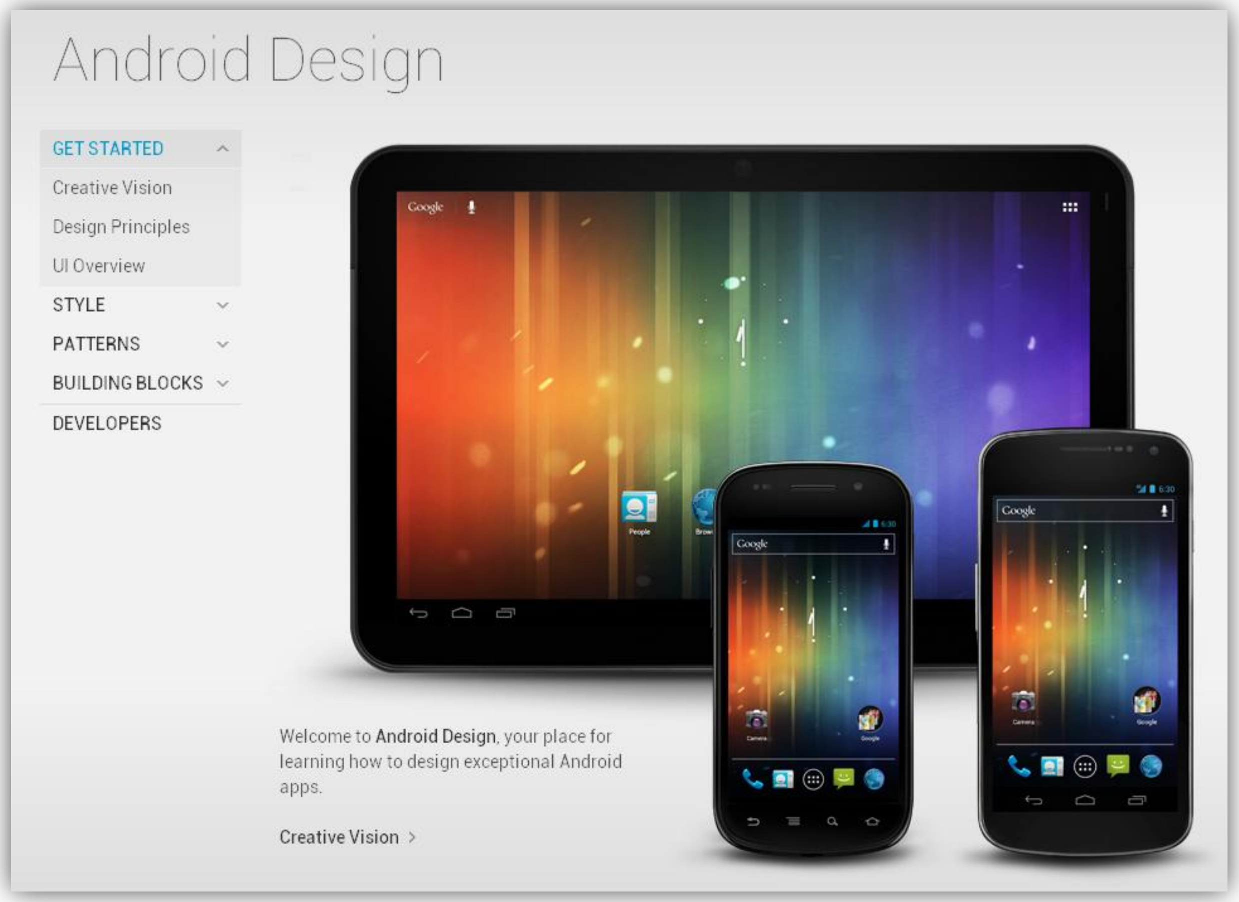


# Extra graphics assets with proper program structure can resolve density challenge

| Android Devices | Device Dedicated Apps | Public Apps |
|---|---|---|
| Wallpaper, boot-up animation, etc. | App launcher icon, menu icon, background images, etc. | |
| To design graphic assets against device physical pixels as usual | | To assume **dp/mdpi** in conceptual design<br><br>To design four versions graphic assets (3:4:6:8 scaling ratio) for **ldpi**, **mdpi**, **hdpi** and **xhdpi** |

**New**

# New Android Design site is official reference

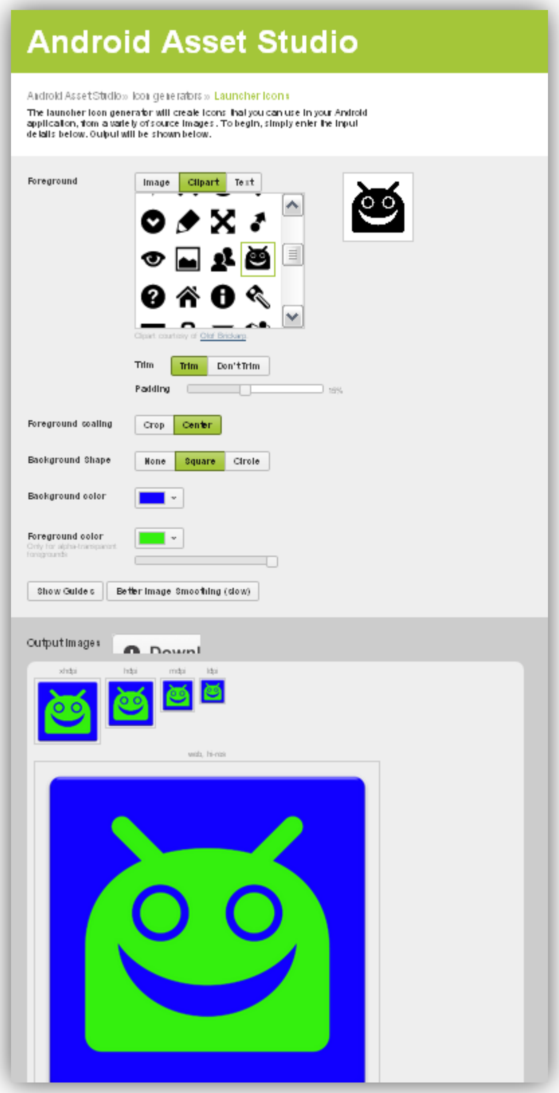


Welcome to **Android Design**, your place for learning how to design exceptional Android apps.

http://developer.android.com/design/index.html

Also can download official Android icon templates pack here
http://developer.android.com/guide/practices/ui_guidelines/icon_design.html#templatespack

## Android Asset Studio can saving your time


Android Asset Studio

Android Asset Studio » Icon generators » Launcher Icons

The launcher icon generator will create icons that you can use in your Android application, from a variety of source images. To begin, simply enter the input details below. Output will be shown below.

Foreground: Image [✓ Clipart] Text

Clipart courtesy of Olof Brickarp

Trim: [✓ Trim] Don't Trim

Padding: 15%

Foreground scaling: Crop [✓ Center]

Background Shape: None [✓ Square] Circle

Background color: 🟦

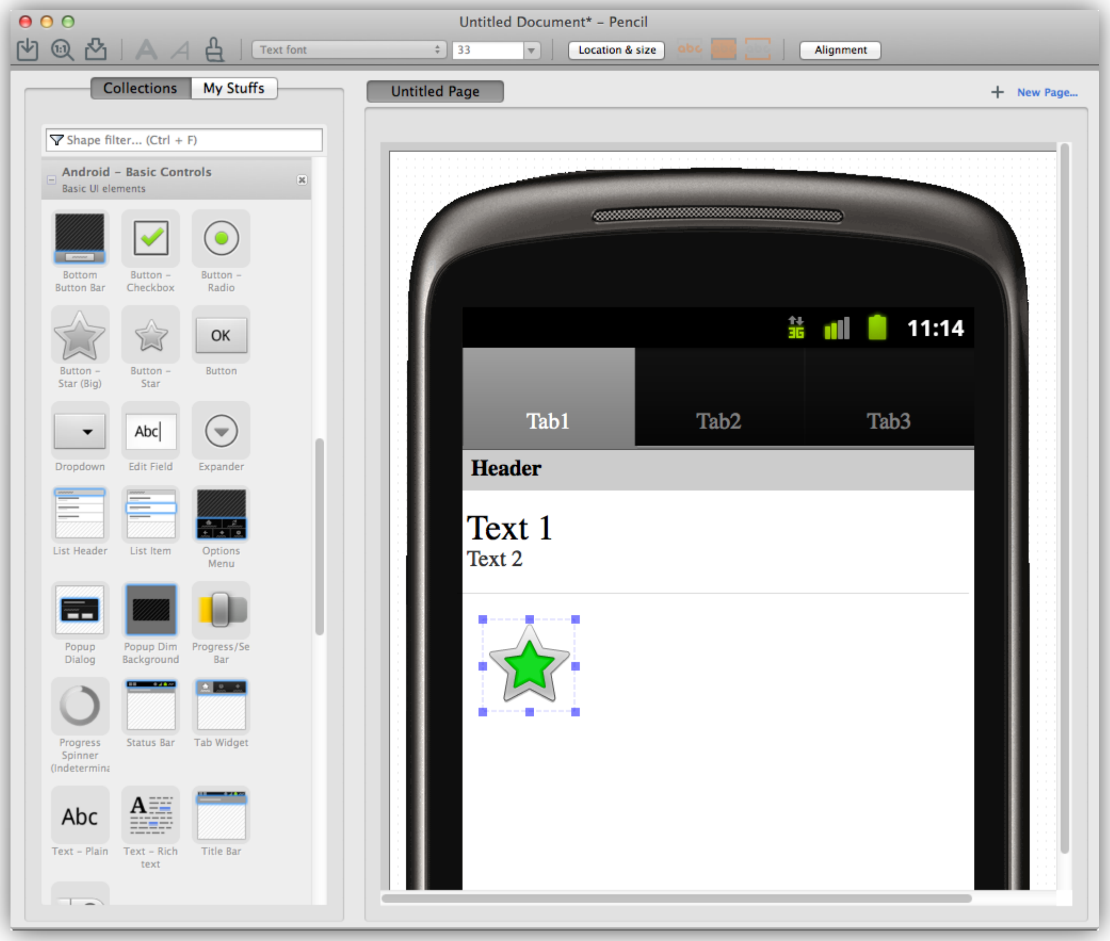
Foreground color: 🟩
Only for alpha-transparent foregrounds

Show Guides | Better Image Smoothing (slow)

Output images

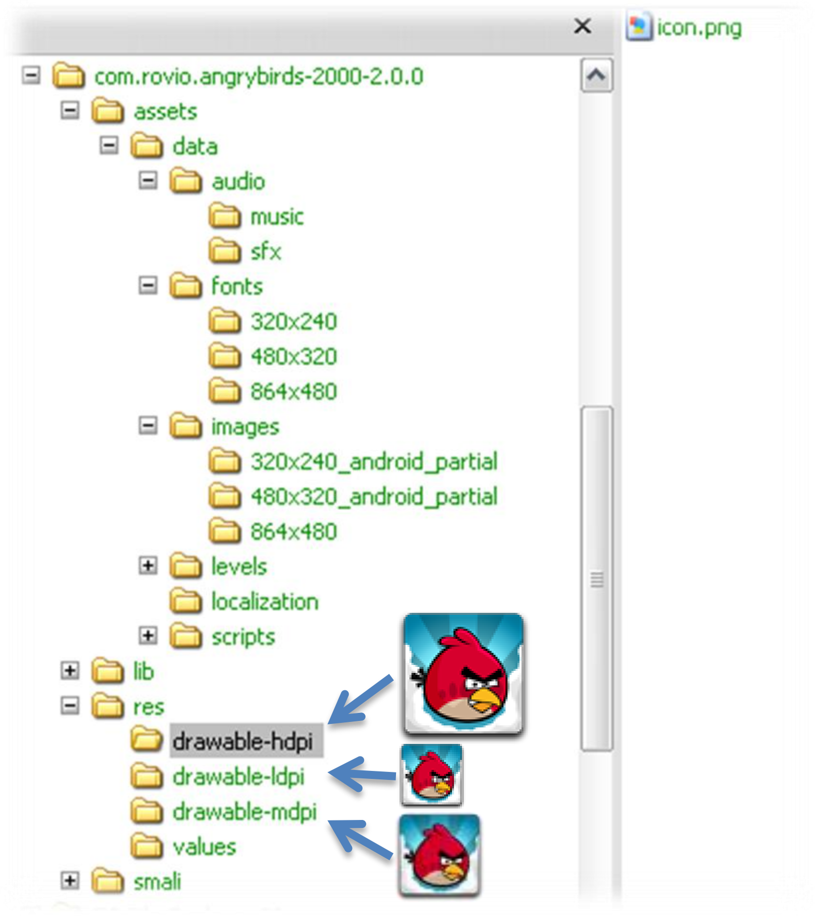
http://j.mp/androidassetstudio

# Pencil with Android Stencils is a nice UI prototyping tool



http://pencil.evolus.vn
http://code.google.com/p/android-ui-utils/

# Key takeaway - public Android apps request more graphics assets



```
icon.png
com.rovio.angrybirds-2000-2.0.0
  assets
    data
      audio
        music
        sfx
      fonts
        320x240
        480x320
        864x480
      images
        320x240_android_partial
        480x320_android_partial
        864x480
      levels
      localization
      scripts
  lib
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    values
  smali
```

Case study for Angry Bird

- To assume **dp/mdpi** in conceptual design
- To design four versions graphic assets (3:4:6:8 scaling ratio) for **ldpi**, **mdpi**, **hdpi** and **xhdpi**

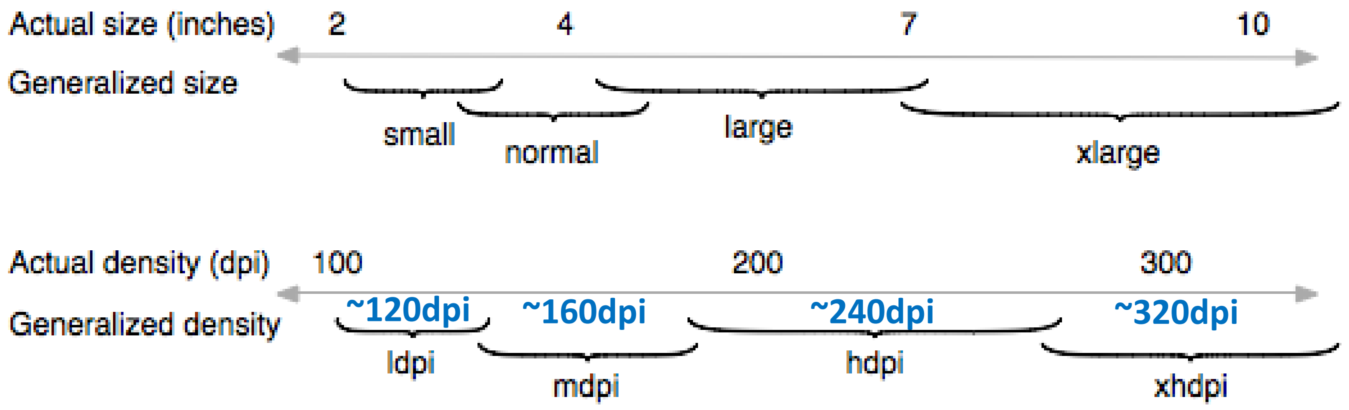
## Technical details …

## Technical terms and concepts

- Density-independent pixel (**dp**)
  - A **virtual pixel unit** that you should use when defining UI layout, to express layout dimensions or position in a density-independent way.
  - The density-independent pixel is equivalent to one physical pixel on a 160 dpi screen, which is the baseline density assumed by the system for a "medium" density screen. At runtime, the system transparently handles any scaling of the dp units, as necessary, based on the actual density of the screen in use. The conversion of dp units to screen pixels is simple: **px = dp * (dpi / 160)**. For example, on a 240 dpi screen, 1 dp equals 1.5 physical pixels. You should always use dp units when defining your application's UI, to ensure proper display of your UI on screens with different densities.

# Technical terms and concepts $^{\text{cont.}}$

- Screen size
  - Actual physical size, measured as the screen's diagonal. For simplicity, Android groups all actual screen sizes into four generalized sizes: **small**, **normal**, **large**, and **extra large**.
- Screen density
  - The quantity of pixels within a physical area of the screen; usually referred to as **dpi** (dots per inch). For simplicity, Android groups all actual screen densities into four generalized densities: **low**, **medium**, **high**, and **extra high**.

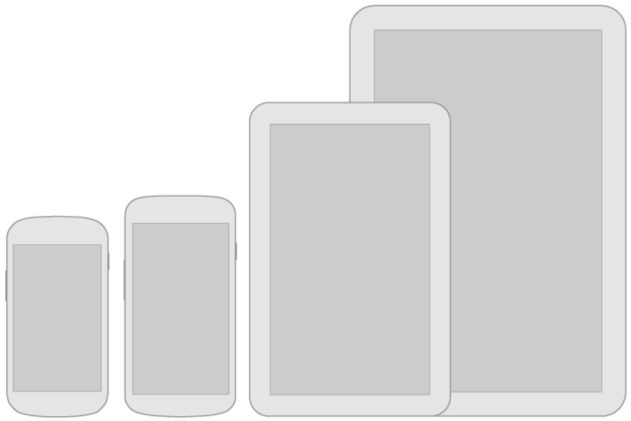


## Technical terms and concepts $^{\text{cont.}}$

- Orientation
  - The orientation of the screen from the user's point of view. This is either **landscape** or **portrait**, meaning that the screen's aspect ratio is either wide or tall, respectively. Be aware that not only do different devices operate in different orientations by default, but the orientation can change at runtime when the user rotates the device.
- Resolution
  - The total number of physical **pixels** on a screen. When adding support for multiple screens, applications **do not work directly with resolution**; applications should be concerned only with screen size and density, as specified by the generalized size and density groups.

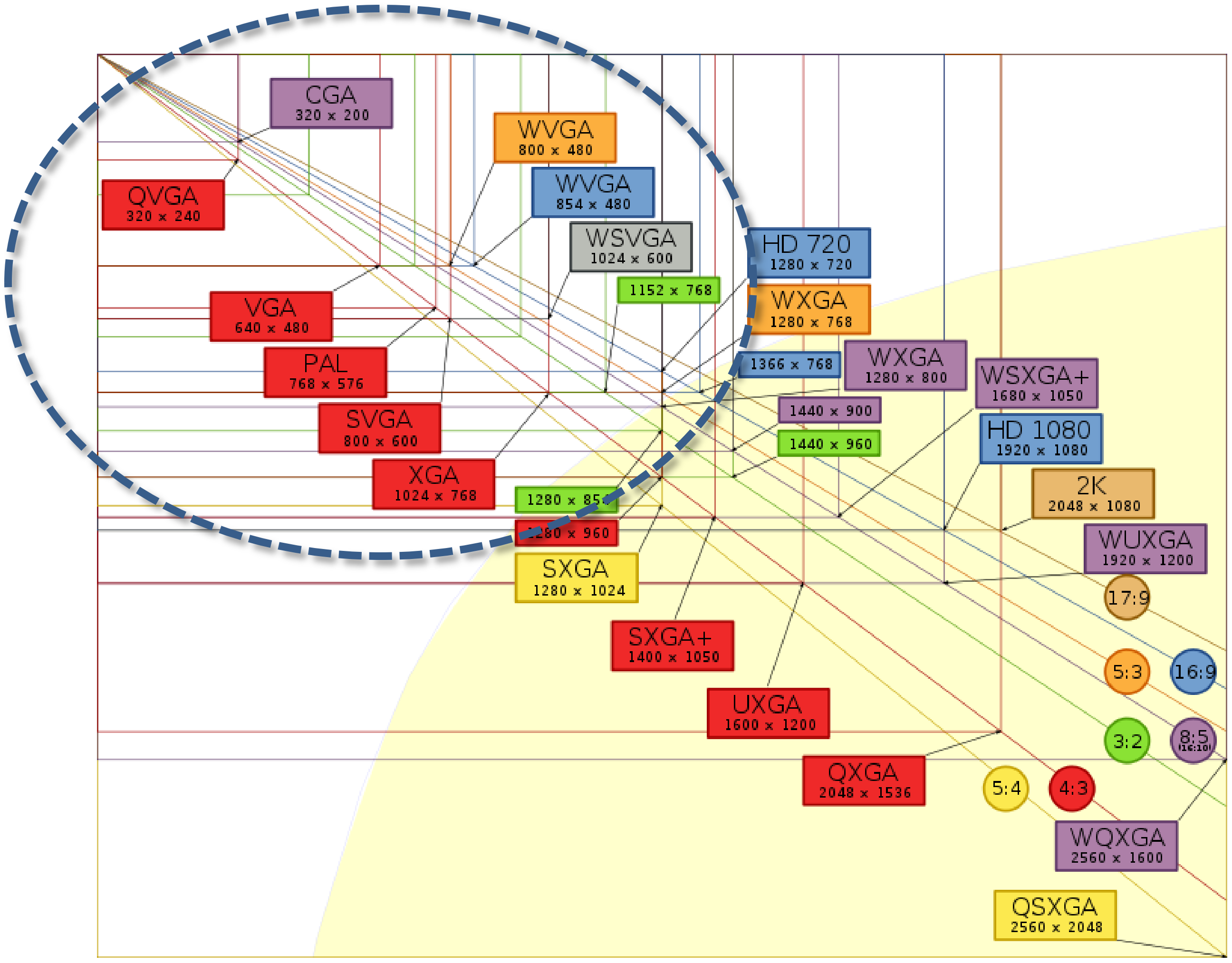
## Range of screens supported

| | |
|---|---|
| **xlarge** screens | > 960 dp x 720 dp |
| **large** screens | > 640 dp x 480 dp |
| **normal** screens | > 470 dp x 320 dp |
| **small** screens | > 426 dp x 320 dp |

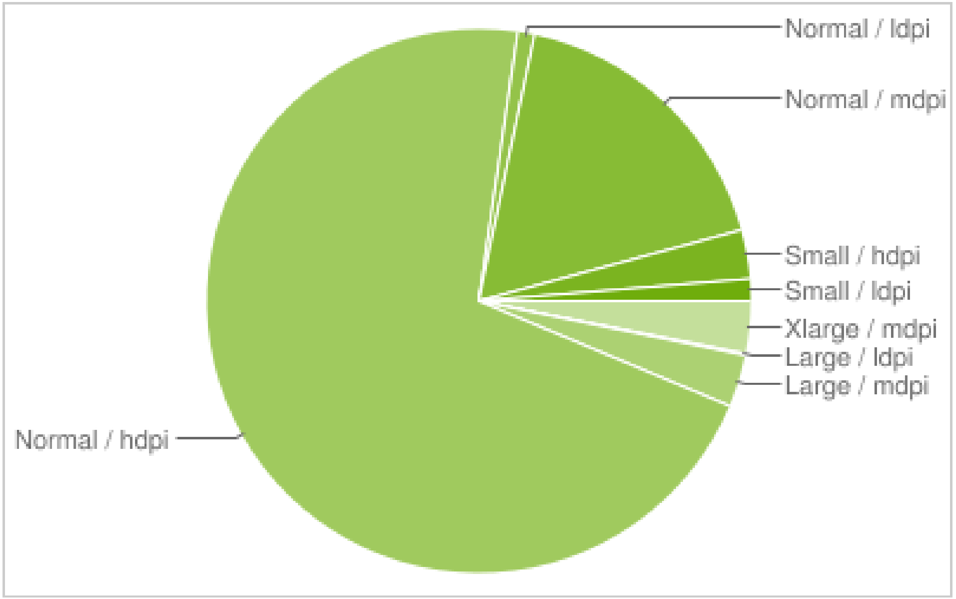


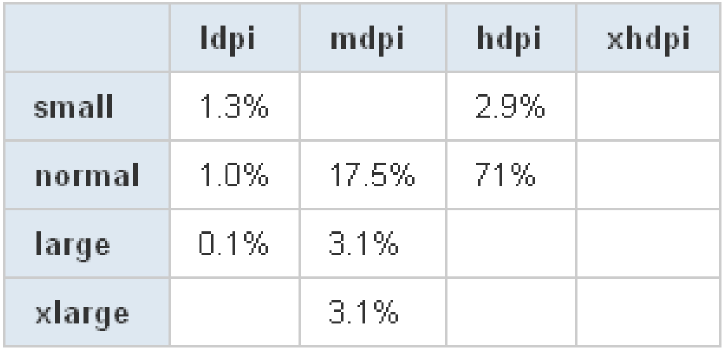
- Note: The system bar in Android 3 and above reduces app space

## Range of screens supported cont.



## Range of screens supported cont.





| | ldpi | mdpi | hdpi | xhdpi |
|---|---|---|---|---|
| small | 1.3% | | 2.9% | |
| normal | 1.0% | 17.5% | 71% | |
| large | 0.1% | 3.1% | | |
| xlarge | | 3.1% | | |

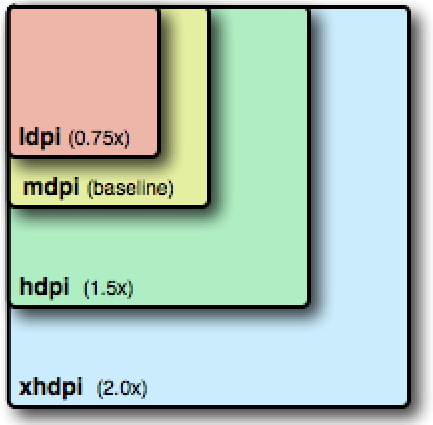
*Data collected during a 7-day period ending on December 1, 2011*

## Alternative graphics assets for different screen densities

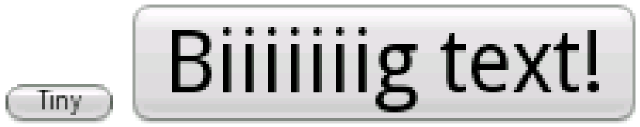
- The configuration qualifiers you can use for density-specific resources are **ldpi** (low), **mdpi** (medium), **hdpi** (high), and **xhdpi** (extra high). For example, bitmaps for high-density screens should go in drawable-hdpi/.
- By default, Android scales your **bitmap drawables** (.png, .jpg, and .gif files) and **Nine-Patch drawables** (.9.png files) so that they render at the appropriate physical size on each device. For example, if your application provides bitmap drawables only for the baseline, medium screen density (mdpi), then the system scales them up when on a high-density screen, and scales them down when on a low-density screen. This scaling can cause artifacts in the bitmaps. To ensure your bitmaps look their best, you should include **alternative versions** at different resolutions for different screen densities.

## Alternative graphics assets for different screen densities cont.

- To create alternative bitmap drawables for different densities, you should follow the **3:4:6:8** scaling ratio between the four generalized densities. For example, if you have a bitmap drawable that's 48x48 pixels for medium-density screen (the size for a launcher icon), all the different sizes should be:
  – 36x36 for low-density
  – 48x48 for medium-density
  – 72x72 for high-density
  – 96x96 for extra high-density

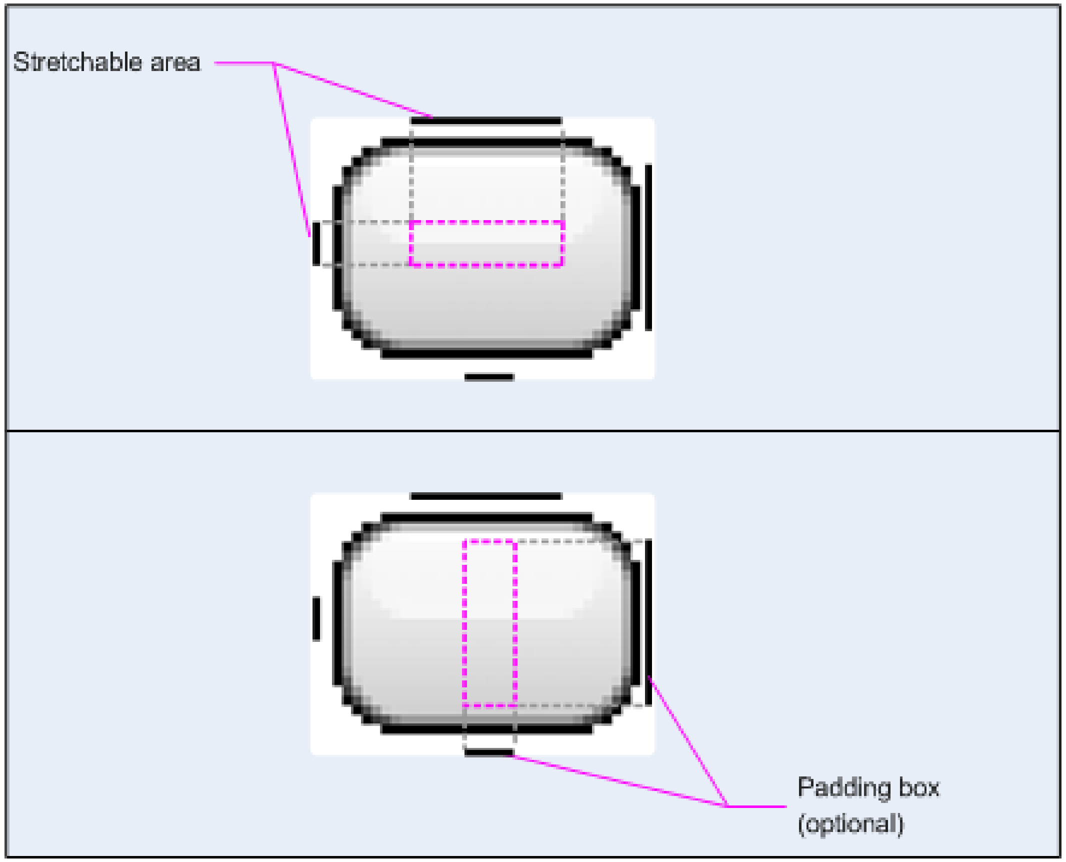


## Nine-patch graphics assets

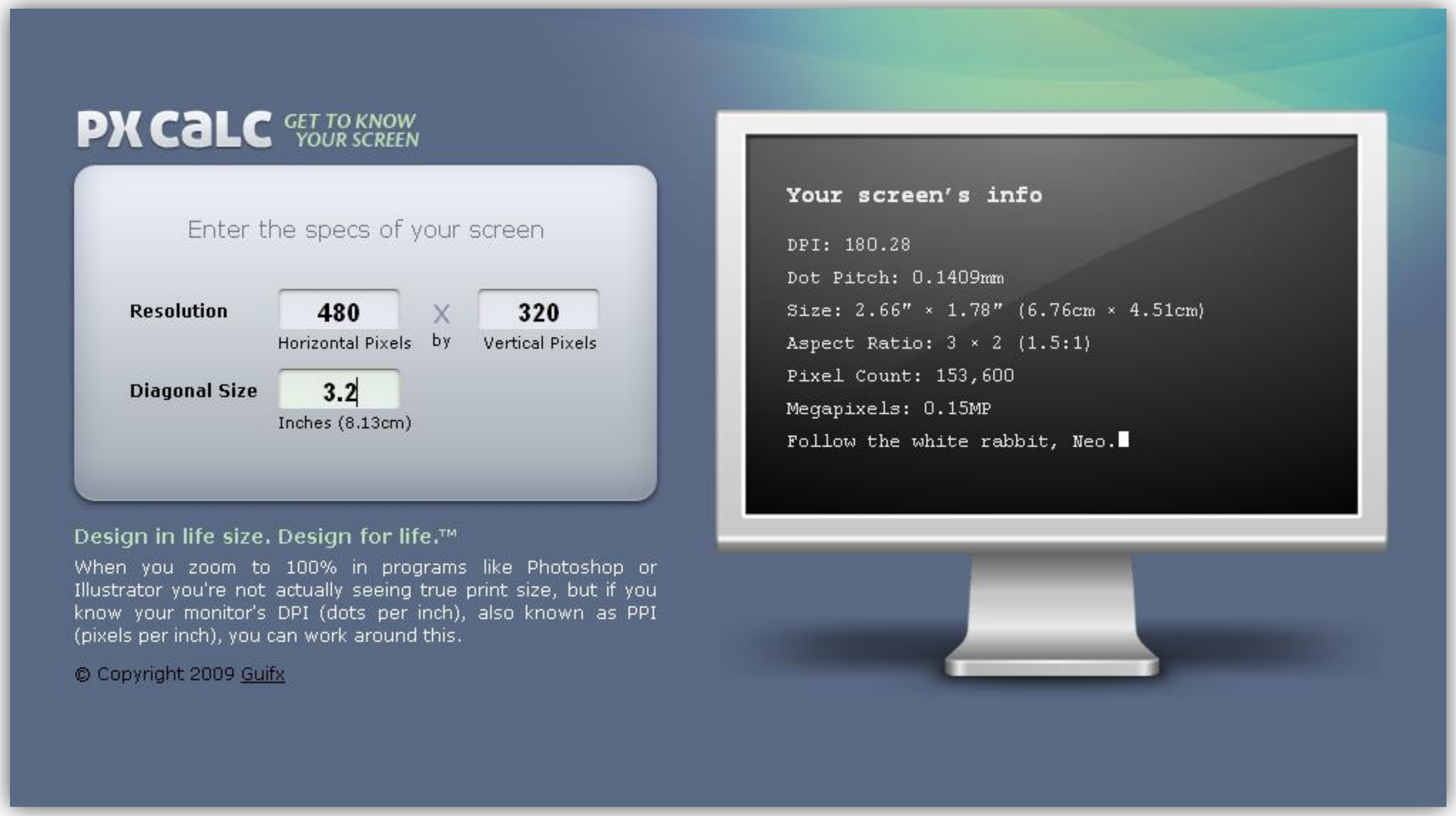


- A **NinePatchDrawable** graphic is a stretchable bitmap image, which Android will **automatically resize** to accommodate the contents of the View in which you have placed it as the background. An example use of a NinePatch is the backgrounds used by standard Android buttons — buttons must stretch to accommodate strings of various lengths. A NinePatch drawable is a standard PNG image that includes an extra 1-pixel-wide border. It must be saved with the extension .9.png, and saved into the res/drawable/ directory of your project.

## Free Android tool: SDK/tools/draw9patch



# DPI online calculator


PX CALC GET TO KNOW YOUR SCREEN
Enter the specs of your screen

Resolution 480 Horizontal Pixels X by 320 Vertical Pixels

Diagonal Size 3.2 Inches (8.13cm)

Design in life size. Design for life.™

When you zoom to 100% in programs like Photoshop or Illustrator you're not actually seeing true print size, but if you know your monitor's DPI (dots per inch), also known as PPI (pixels per inch), you can work around this.

© Copyright 2009 Guifx

```
Your screen's info

DPI: 180.28
Dot Pitch: 0.1409mm
Size: 2.66" × 1.78" (6.76cm × 4.51cm)
Aspect Ratio: 3 × 2 (1.5:1)
Pixel Count: 153,600
Megapixels: 0.15MP
Follow the white rabbit, Neo.
```

http://www.pxcalc.com/

## Technical details for supporting multiple screens

- http://developer.android.com/guide/practices/screens_support.html