Charles Sturt University

ITC209: Mobile Application Development

Topic 3: GUI Design

Robin GAO

School of Computing and Mathematics

Where to begin….

- There is honestly so much to talk about with regards to User Interface Design it's hard to think about where to begin.
  - Especially for just 1 week

- UI Design really is another topic altogether
  - Especially when we look at how to design for a specific platform/medium

- We want to have an understanding of the following for our projects:
  - Design considerations when looking at mobile applications
  - General UI Design principles
  - Understand the Android design philosophy
  - How it is implemented within a project

UI Design for the Assignment

- Need mock-ups for all the main interfaces that your application has

- A general storyboard of the flow of UI elements
  - Helps identify the activities/classes you will need
  - Format completely up to you
  - Present it in a way that is clear and understandable

- Practice the storyboard – click on your drawings and ‘Act’out how the UI will work
  - Will help identify any major missing flows

Designing UI for Mobile Apps

- One major thing to consider (and especially for Android and UWP):

We don't design for Absolute Positioned Layouts

- Mobile apps adapt to many different screen sizes, resolutions and ***orientations***
  - Additionally...we don't design to pixels (px)…They ignore the notion of physical size

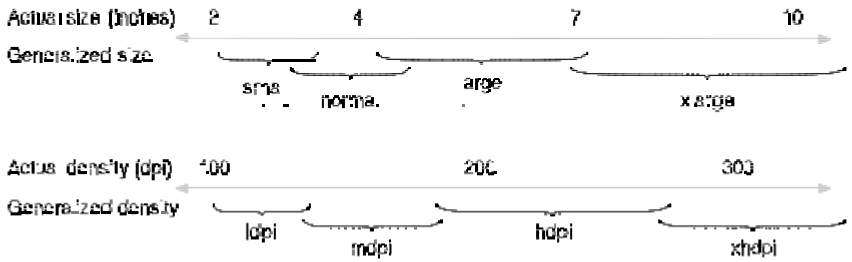
- The number of pixels within the physical area is referred to as the screen density
  - Or dots per inch (dpi)

- Density Independent Pixels (dp/dip) are virtual pixels that scale up based on the dpi of the screen

Screen Density

- Android groups devices based on dpi – density buckets
  - Auto assigns to the closest match – and helps control scaling.

Density	DPI
ldip (low)	120
mdpi (medium)	160
hdpi (high)	240
xhdpi (extra-high)	320
xxhdpi (extra-extra-high)	480
xxxhdpi (extra-extra-extra-high)	640



Source: https://developer.android.com/guide/practices/screens_support.html

UI Design Principles

1. *Focus on the User*

- If the user is not happy with what they see, they won't use it
- Design with a user-first mentality - step into their shoes
  - Know the user – *BE the user*
  - *And remember – the user is* ***reallllllllly*** *stupid*
- Users want a simple and fast experience
  - Don't overdo modal progress and confirmation dialogs

UI Design Principles

2. Be Predictable and Consistent

- Navigation shouldn't require intense cognitive concentration
  - Reduce the number of actions required
- They expect apps to work like the apps they've used before
  - User have used interfaces before – conform to them – ***copy them***
- Have a consistent style and theme
  - Icons, buttons, links, fonts, formatting should all look the same throughout the app
- Handle orientation changes

UI Design Principles Cont.

3. Make the Right Thing Visible

- The Launch page needs to connect instantly - Most common operations should be immediately visible and available
  - Hide non-relevant content
- If something can be clicked make sure it looks like it
  - Large obvious tap targets with colours that pop out
  - Use Contrasting Colours – Foreground and Background is visible and separate
- The Launch page needs to connect instantly

UI Design Principles Cont.

4. Show Proper Feedback

- Show progress – don't leave the user in the dark
  - Show the spinning loading icon, notify when an action didn't work/an error occurs

- Remember buttons contain different states:
  - Default, Disabled, Hover and Pressed
  - In Touch Interfaces – **no Hover** – but we do have *Focused*
  - Highlights things we are directly using or 'hold' presses

UI Design Principles Cont.

5. *Be Fault Tolerant*

- Constrain possible operations to only those that make sense
  - If a user can't submit a query – disable the ability to do so

- ***If an error is possible someone will make it***
  - This can be prevent through extensive use-case testing and random exploration and experimentation – And common sense.

6. *A Smaller Cog in a Larger Machine*

- Remember your app is sharing resources with the phone's OS
  - What is your app going to do when the phone receives a call?
  - What is your app going to do when the user moves it between the foreground and background?

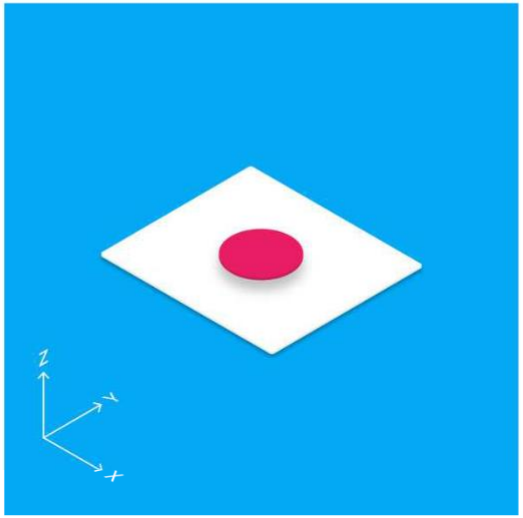
Designing UI for Android

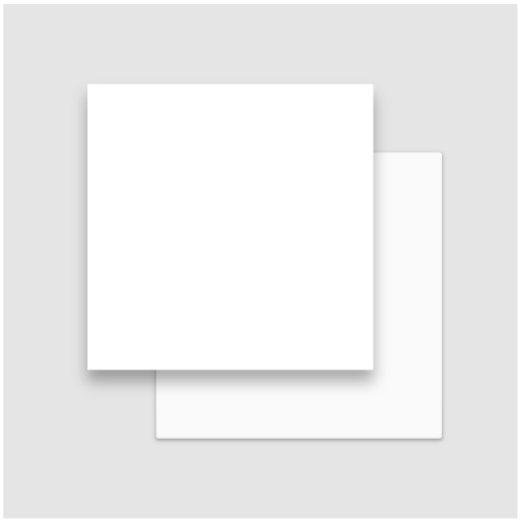
Android Design Principles:

- Designing with Real Objects – Material Design
- Simplicity and Consistency
  - Keep it brief and to the point
  - Only show what needs to be shown
  - Allow for tricks to work across multiple apps
- Clear and Content Rich
  - Constant feedback for all actions
  - Understanding where we are at all times
  - Make important actions stand out

- Material Design – What is it?
  - Essentially making the UI feel like tangible objects
  - Giving weight and dimension to enhance tactile 'feel'
    - Focuses on surfaces, edges, ***light*** and movement
  - Inherits from traditional pen and paper design principles
    - Bold, Graphic and Intentional

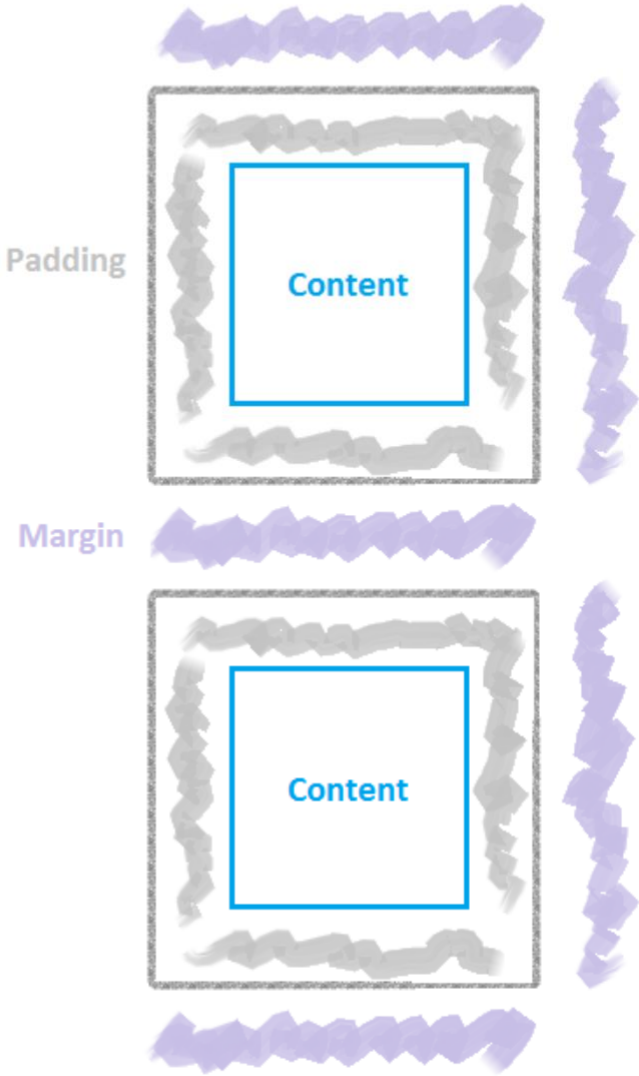
- More here: https://material.google.com/





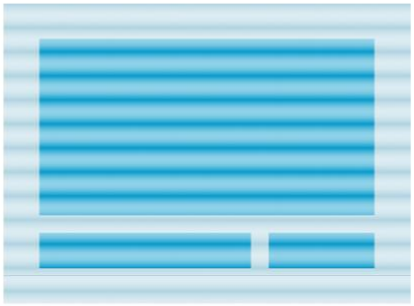
Understanding Layouts

- The Layout .xml works along the same principles of HTML
- Can think of everything enclosed in a box – Box Model
  - This includes the Content + Padding
  - Android does not have borders
- Padding vs Margins
  - Padding is part of the object
    - (Protects contents – packing peanuts)
  - Margins are outside (not part of) the Object
    - Space between objects
  - Margins are between objects
- Each element in the View is structured in a Nested Tree
  - It inherits its positions and scales from its parents

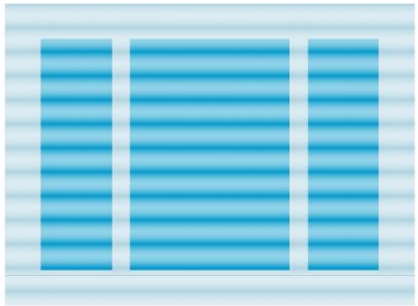


Layouts in Android

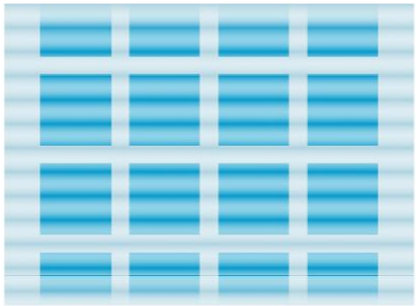
- Layouts Types determine how elements interact and are positioned



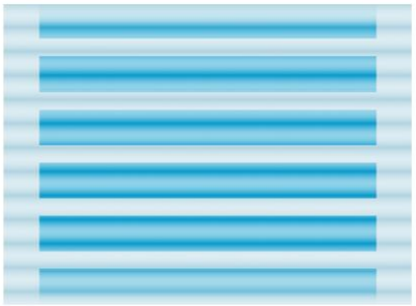
Relative



Linear

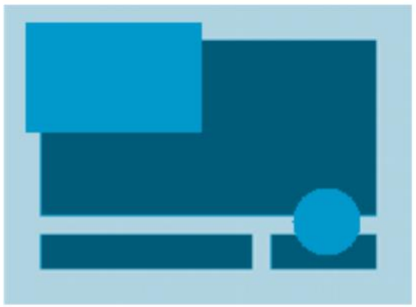


Grid

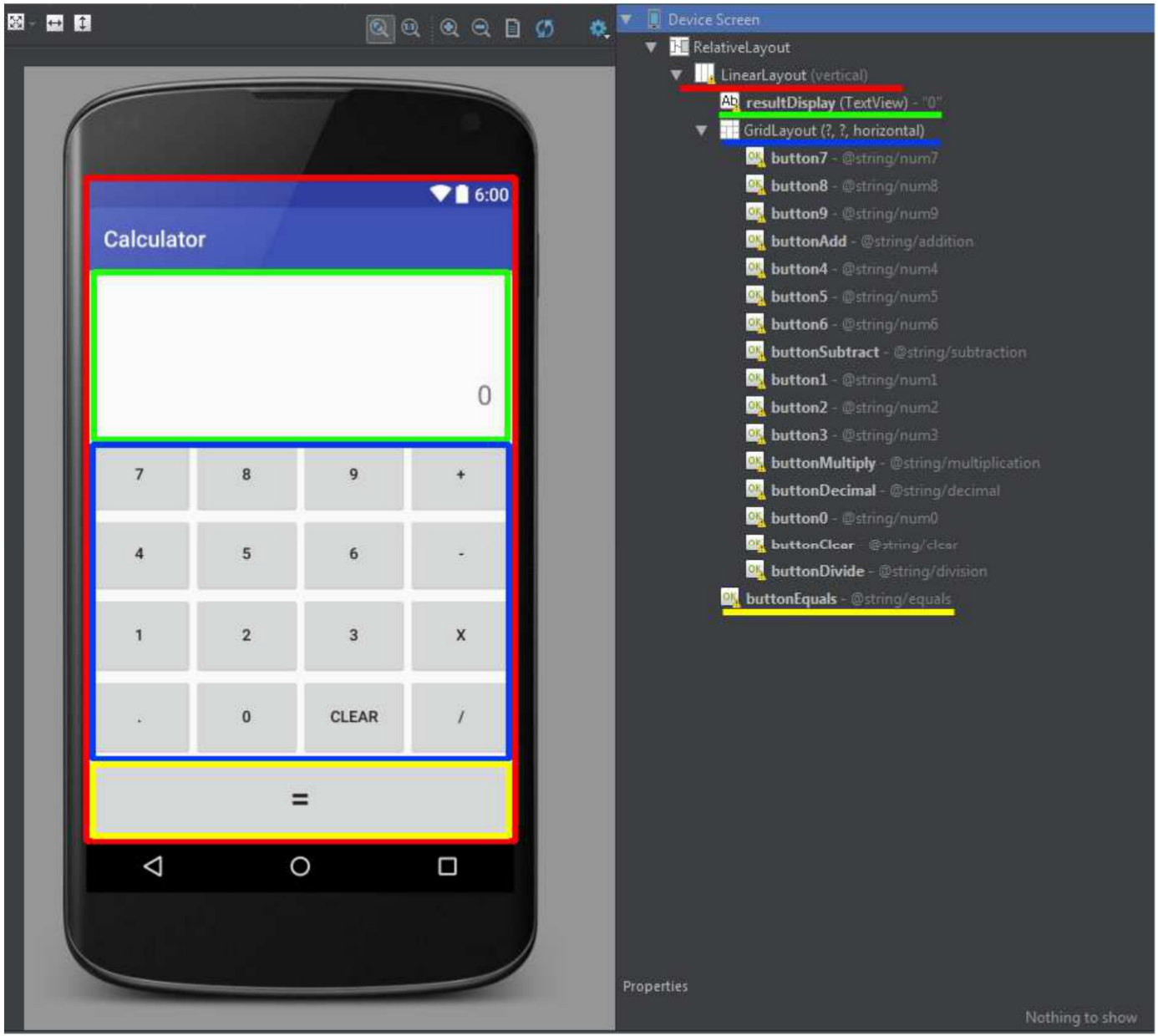


List

Web

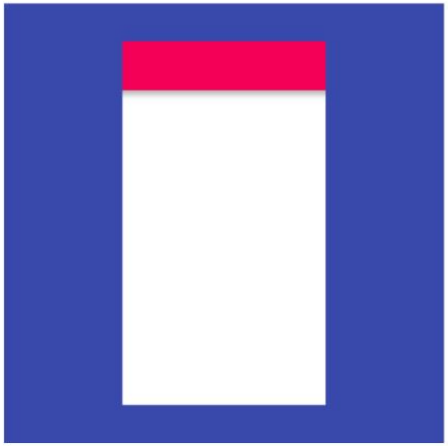


Frame



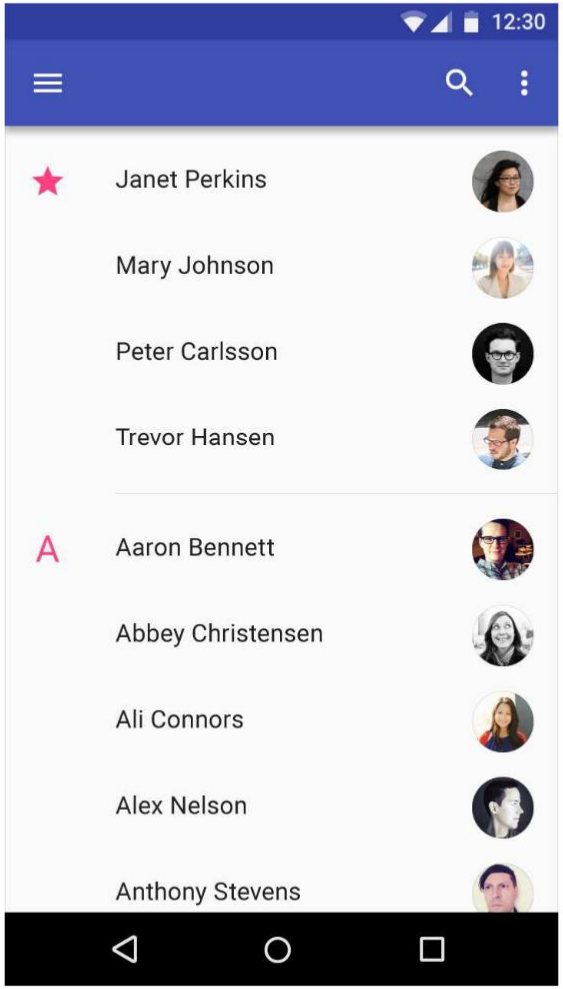
- Width = 1
  - Take up the whole screen
- Height = 0
  - Not sure how large so don't set anything
- Weight
  - Set the w/h to a proportion of the screen
- Elevation
  - Determines shadow and outline of element
  - https://material.google.com/what-is-material/elevation-shadows.html#elevation-shadows-elevation-android-

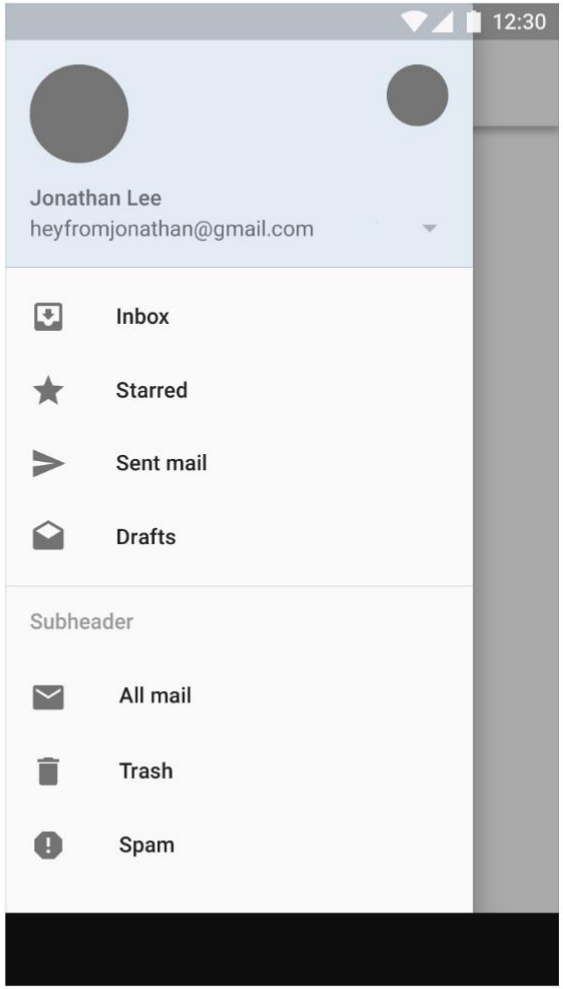
Common Android Components



Toolbars

Navigation Drawer





Colours, Theme and Style

- Google highly recommends giving your App a custom theme/colour
  - To help identify what your app is from
  - others Help user feel at home
- Google provides colour Swatches to use
  - Pick a Primary and an Accent Colour
  - https://material.google.com/style/color.html#color-color-palette
- Styles and Themes work very similar to CSS
  - Provide a Style Sheet for how specific elements should look
- A Style changes the appearance of a single UI element while a Theme does it to a collection of elements
  - Located in Styles.xml
  - https://developer.android.com/training/material/theme.html