Lecture 5 – Android activities and UI

CS260 – Android App Development

Paul Cao

Review

- Basic hello world project on Android
- Java basics

Plan for today

- Mobile development idea
- Android app components
- UI introduction

Mobile development environment

- Pro
  - Small, mobile
  - Popular amongst professionals
  - Can make money on apps

Mobile development environment

- Cons
  - Low processing power
  - Limited RAM and storage
  - Small screen with limited resolution
  - Slow data transfer
  - Battery life

Resource Constraints

- Usually Manufactures focus on larger screens and dimensions of the device
  - CPU power takes a backseat
  - Need more efficient code
- Limited capacity on flash storage
  - Be careful on storing your application data
  - Clean up after yourself

Resource Constraints

- Design for different screens
  - GUI is a must
  - It's a zoo out there wrt screen size.
- Expect low network speed / high latency
  - It is getting better with faster 4G/3G networks
  - Still have to consider the worst case scenario

Android development guidelines

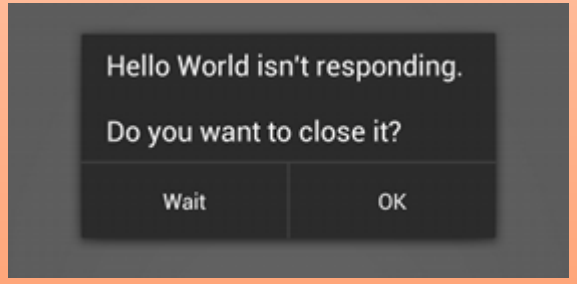
- Be aware of the costs
  - Transfer data as little as possible
  - Caching data/Geocoding results to avoid redundant lookups
  - Schedule big updates on off-peak times or only when connected with Wifi
  - Respect user's preference

Android development guidelines

- Performance
  - A mobile device has about 200M SDRAM
  - Some suggestions (may be contrary to what you learned in programming)
    - Use public fields instead of private fields
    - Avoid double and use float. Use int if you can
    - Use library class/methods (they are written as assembly codes!)
    - Avoid 2D arrays. Use 1D instead
    - More on http://developer.android.com/training/articles/perf-tips.html

Responsiveness

- Your app should never keep user waiting
  - An app must respond to user action within 5 seconds
  - A broadcast receiver must return from its onReceive handler within 10 seconds
  - Or else Delvik VM kills your app.



Data freshness and secure application

- Multi-tasking means your current data may not be fresh
  - Need to think about how/when to refresh your data
  - Usually use a background service for that
- Security is important
  - Require permission for any broadcast
  - Be careful when you application accepts input from external sources (e.g. internet, bluetooth, etc)
  - Minimize the permission your app requires

One last word

Your app is probably not the most important software on the user's device

1. Phone
2. SMS
3. Email
4. Mp3
5. Your app

What makes an Android Application?

- Android components are loosely connected
  - Bound by the manifest
- Activities → the presentation layer (face)
- Services → invisible workers of your
- Content providers → sharable persistent data storage
- Intent → inter-app message-passing framework
- Broadcast receivers → Let your app respond to other app's intent
- Notifications → Let you know subtly

The manifest

- It defines the structure and metadata of your app, its components, and requirements
- A manifest is an xml file

<manifest>

<nodes>.....</nodes>

…

...

</manifest>

Node types in the manifest

- uses-sdk
- uses-configuration
- uses-feature
- supports-screens
- uses-permission
- application

Creating resources

- Idea: separate resources from the codes
- string
- color
- plurals
- string-array
- array
- dimen
- layout

- And more (animation, menu, style)

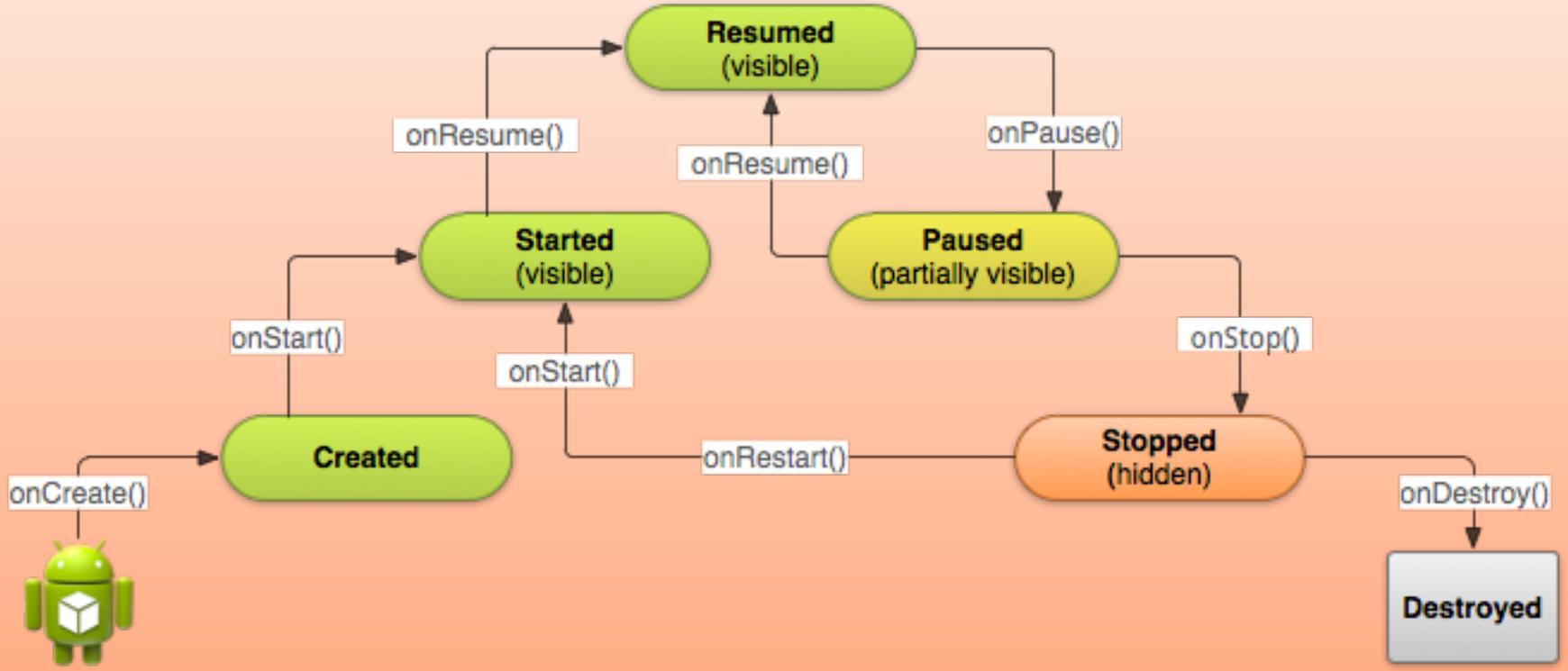
Creating resources

- Most of time, each resource should have a unique name
  - They can be used in your layout via @type/name
    - @string/hello_world
    - @color/red
  - They can be used from getResources() in your code.
    - Resources resource=getResources();
    - Int blue=resource.getColor(R.color.special_blue);

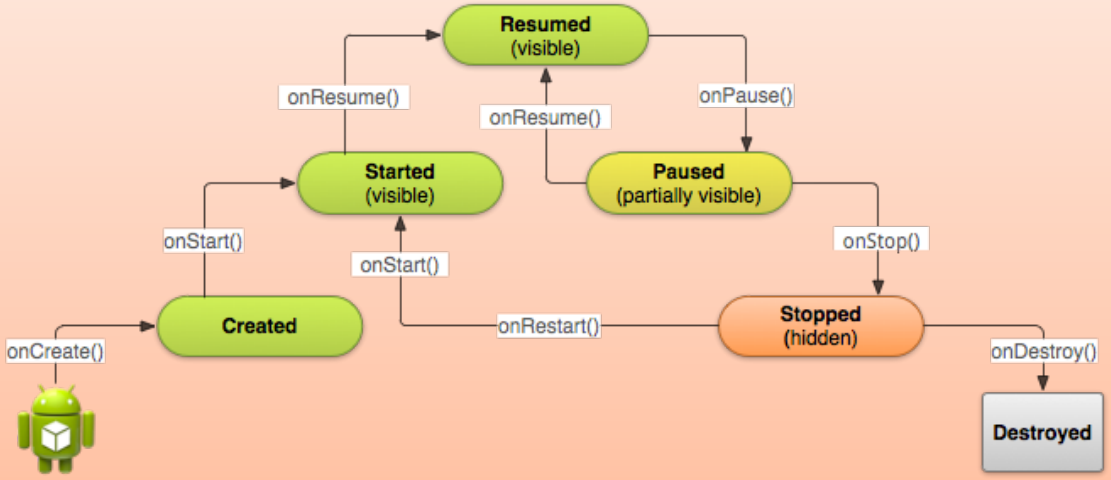
Activity in Android App

- Activities are basically the user interfaces of your app
- Each screen is a different activity
- Most of the activities rely on your layout
  - Android provides a big list of View groups
    - All of them are child of the View class

Activity cycles in Android (callback methods)

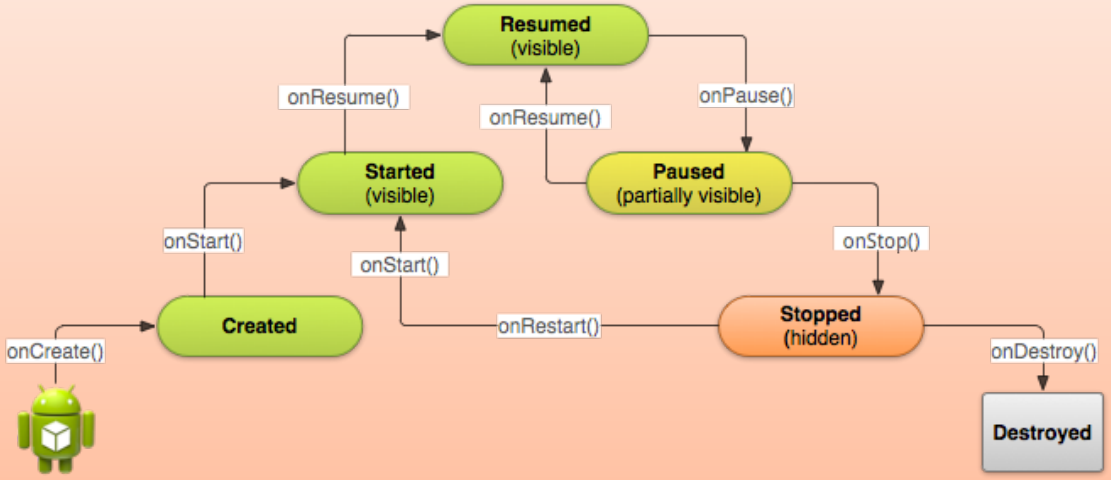


Activity cycles in Android (callback methods)



- Only Resumed, Paused, and Stopped are permanent states
- Created and Started are transient states
- It is important to save your data when transiting into paused or stopped state

Activity cycles in Android (callback methods)



- Visible lifetime: from onStart() to onStop()
- Active lifetime: Resumed and Paused

UI design

As if a device can function if it has no style. As if a device can be called stylish that does not function superbly.... Yes, beauty matters. Boy, does it matter. It is not surface, it is not an extra, it is the thing itself.

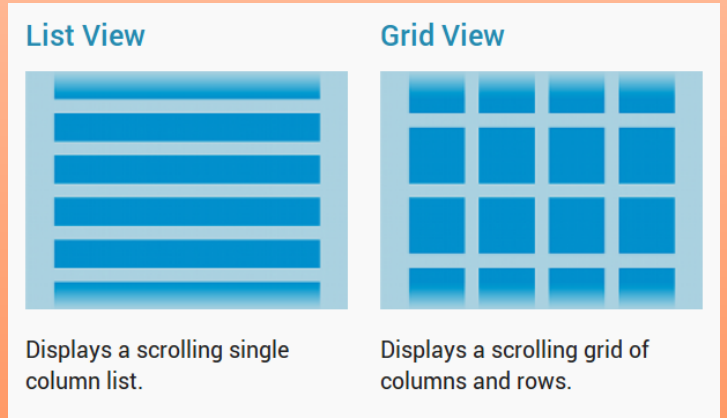
STEPHEN FRY , THE GUARDIAN (OCTOBER 27, 2007)

UI basics

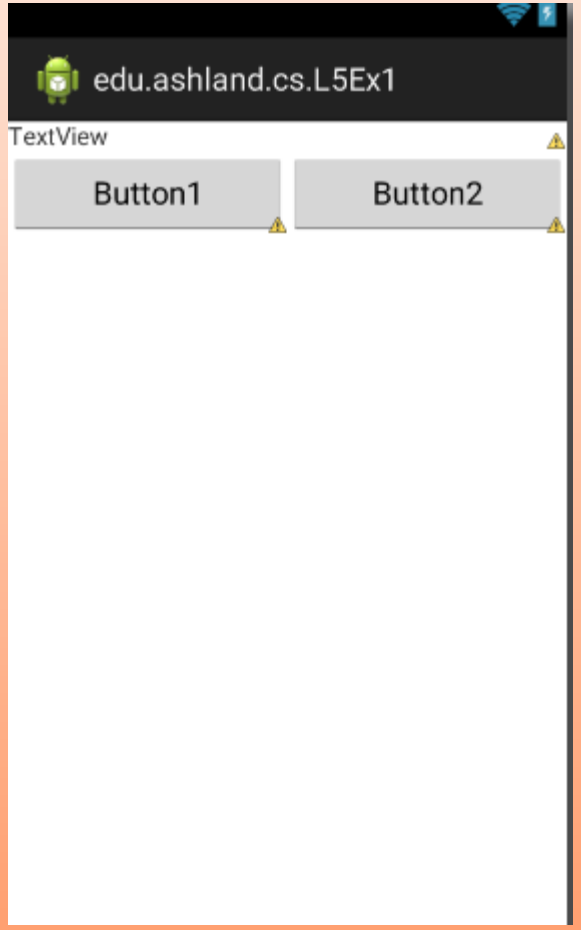
- View: the basic class that every widget inherited from
- Fragments: A group of basic views encapsuled into an activity. → to accommodate different sizes of screens
- Activity uses setContentView() to create UI
  - Usually setContentView() is called inside onCreate()
  - SetContentView uses either a layout or a View as its parameter

Layout basics

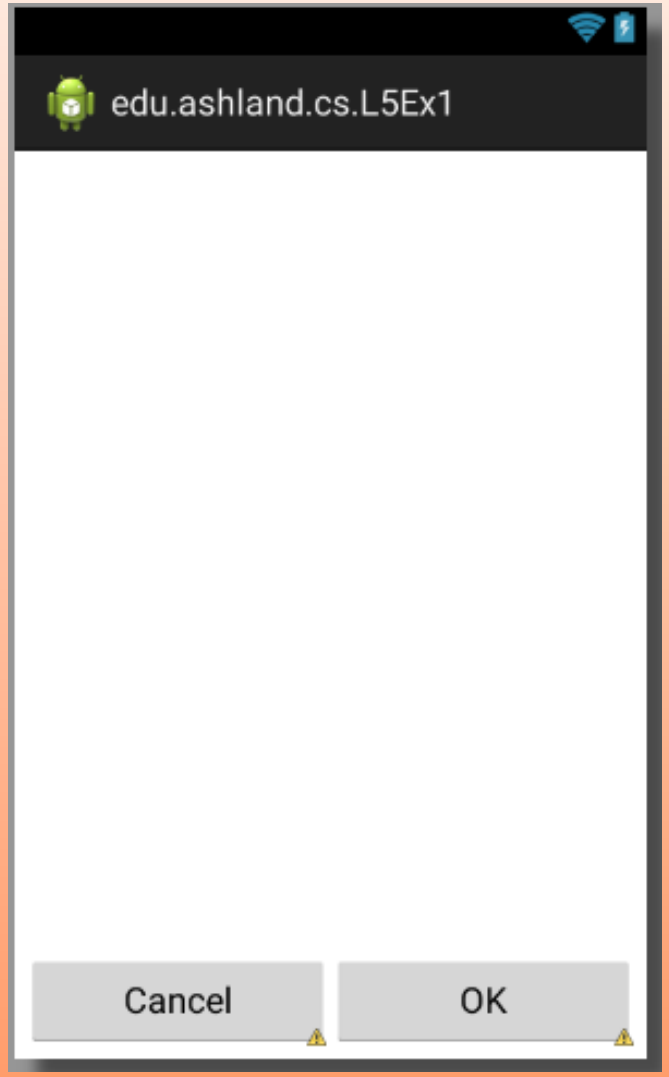
- http://developer.android.com/guide/topics/ui/declaring-layout.html
  - Linear layout: vertical or horizontal linear layout of all the objects. One after another
  - Relative layout: layout your Views into relative positions from each other
  - Grid/List layout: new to Android and uses Adapters to populate list of items into the layout.



Linear Layout Exercise

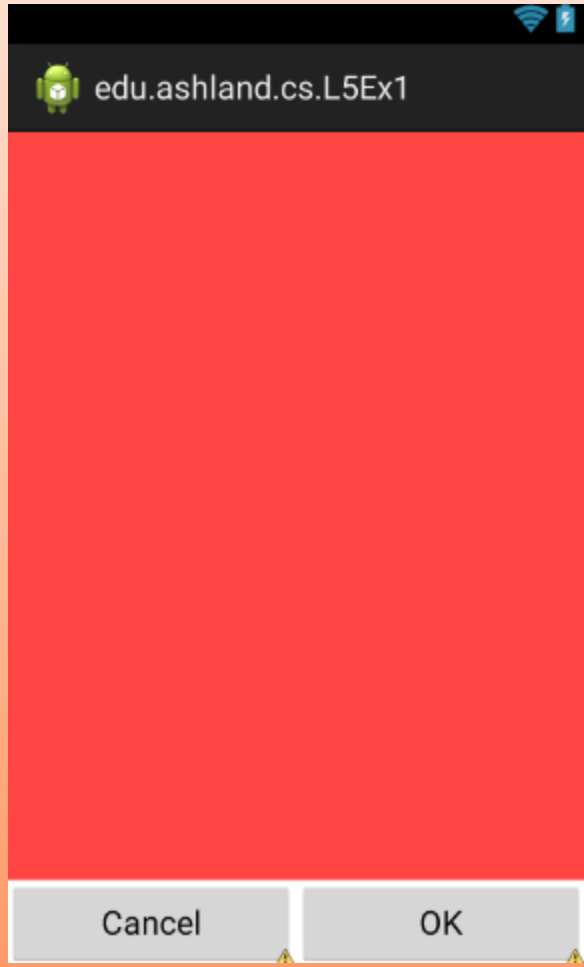


Relative layout



Grid layout

- Note: no layout_width and layout_height aren't required for grid_layout elements



Example

Create a AU pick color app.

Exercise

Create a palindrome app