Android Lesson 1 Questions
Why did you choose to devote a significant part of the course to Android?

Android incorporates many important ideas of modern software

- Object-oriented frameworks
- Excellent use of design patterns
- GUI / Event-driven programming
- Responsive design
- Modern tools
- Networking, databases, ...
- In Java (so I don’t have to teach a new language)
- Give you a concrete skill for internships / summer hacking
Why did Udacity/we choose Android Studio for Android App Development?

Android Studio is the tool that Google produces for Android App development.

What strengths does Android Studio have over other environments (such as IntelliJ, which I believe is a viable IDE for Android App Development)?

Android Studio is based on IntelliJ
Where are we going?

Will we continue to watch these Udacity lessons, or did we just watch the first one as a warmup for getting into Android?

- We will watch a bunch; not sure how far we’ll get, but if you want to watch ahead, it won’t be wasted time.

Are we all going to be programming the same app or is this going to be a more open project?

- Starting with the same app, but will work up to open-ended final project.
Alternatives

Why use Android when you could make something in React Native and then compile to android?

- Native (android) apps should be faster, use less power, and have access to lots of system services/devices. Advantages of React Native is write once run on Android, iOS, and web.

Is it possible to use HTML/CSS to build the front end of an app?

- Not that I know of. Doesn’t have the information needed.
Fragments

The tutorial comes with code such as automatically generated fragment as an inner class. However, isn't this outdated? So we won't be using the exact same style in class.

What does fragment_main.xml really do? My android studio didn't have it to begin with, so I had to create one manually. I don't get what it does, though.

I'd like to see an example of multiple views sharing fragments.

- Fragments aren’t an outdated idea, and you can get Android Studio to generate a default fragment for you either through selection of a BasicActivity or the Master/Detail Flow. Both can be found through: New->Activity
Getting Sunshine to work

The starter code required obtaining an API key.

Kept getting a "cannot resolve method findViewByID" and "cannot resolve symbol listview_forecast" on the last step.

- The findViewByID method is implemented both for Activities and Views. A Fragment is neither; need to get a View.

In OnCreateView:
View rootView = inflater.inflate(R.layout.fragment_detail, container, false);
... = rootView.findViewById(...);

- Ensure a component: android:id="@+id/listview_forecast"
Human Factors / User Interface Design Guidelines
How would you interact with this?

First Launch Date: 09/09/97  Set Date
First Launch Time: 19:17  Set Time

Constrain input
Simplicity
Signifiers
How about this?
Design Terminology: Affordances

- An affordance is a relation between an object/environment and an organism that affords the opportunity for that organism to perform an action.
- For example, a knob affords twisting, and perhaps pushing, while a cord affords pulling.
- An affordance enables the possibility of some action.
Design Terminology: Signifiers

- A "signifier" is some sort of indicator, some signal in the physical or social world that can be interpreted meaningfully
User Experience design in a nutshell

- Provide the affordances desired by users

- Organize those affordances appropriately
  - “Information Architecture”

- Make your affordances obvious through correct use of signifiers
User Experience Design is Hard

- Most users are not like you

- Users can’t always tell you what they want

- But, they can sure tell you what is wrong.
  - Consistent problems are the system’s fault
UX Design is an Iterative Process

- Iteration can be costly
  - Might have to re-write a lot of your code
Spiral Model of Design

- Use throwaway prototypes and cheap evaluation early in the cycle.
Usability

- How well users can use the system.

- Dimensions of Usability:
  - Learnability: Easy to learn?
  - Visibility: Is the state of the system clear?
  - Efficiency: Once learned, is it fast to use?
  - Errors: are errors few and recoverable?
  - Satisfaction: is it enjoyable to use?

- Design Principles guide building usable systems
  - Guided by findings in Ergonomics / Human Factors
Ergonomics / Human Factors

- the scientific discipline concerned with the understanding of interactions among humans and other elements of a system

- applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance

- Characterizes the capabilities and limitations of humans
Human Factors: Power Law of Practice

- The log time to complete a task decreases linearly with the number of practice trials.
Usability

- How well users can use the system.

- Dimensions of Usability:
  - Learnability: Easy to learn?
  - Visibility: Is the state of the system clear?
  - Efficiency: Once learned, is it fast to use?
  - Errors: are errors few and recoverable?
  - Satisfaction: is it enjoyable to use?

- Design Principles guide building usable systems
  - Guided by findings in Ergonomics / Human Factors
What is wrong with this?
Design Principles for Learnability

- Consistency (in design and metaphors)
  - Similar looking things act similarly
  - Different looking things act differently
  - In wording, location, color, ordering

- Use common words not jargon

- Recognition, not recall
  - Labeled buttons rather than command languages

- 2 kinds of users: Beginners & Experts
Alternative Design
Visibility

- Avoid affordances w/o signifiers
  - E.g., pop-up menus by clicking on open space

- Avoid hidden modes

- Perceptual Fusion:
  - Stimuli <100ms apart seems fused to humans
  - 10 frames/second appears as moving picture
Efficiency: Hand motion tasks

- Moving mouse on screen
- Finger on keyboard/touchscreen
- Hand from keyboard to mouse

Fitts’s Law

- Time = $a + b \log(D/S)$
- $\log(D/S) = \text{index of difficulty}$

- Bigger, closer = easier
- For mousing, $S$ is infinite on edge of screen (good!)
Efficiency: Path Steering Tasks

- Fitts’s law only for unconstrained movements
- Task much harder if constrained to tunnel

\[ T = a + b \left( \frac{D}{S} \right) \]
Design for Efficiency

- Fitt’s Law & Steering Law
  - Important targets: big, nearby, or at screen edges
  - Avoid steering tasks

- Provide shortcuts
  - Keyboard accelerators
  - Bookmarks
  - History
Are you sure you want to delete all records from the database?

Yes

No

Green = go, safe

Red = danger, stop
Errors

- Principle: Protect User’s Work

- Prevent Errors as much as possible:
  - Selection rather than typing (within reason)
  - Constrain user input
  - Separate risky commands from common ones

- Undo
- Confirmation Dialogs (when appropriate)
- Understandable error messages
Simplicity

- Less is More
  - Omit extraneous information, graphics, features

- Hick's Law:
  - time to make a decision is proportional to log(# choices)
Whenever your local SMS Administrator sends you an actual software Package, the SMS Package Command Manager will appear (usually at network logon time) displaying the available Package(s). The following screenshots display scenes similar to what you will see when you receive an actual SMS Package.

To start the demonstration, click the "CLICK HERE TO START" button of the screen.
<table>
<thead>
<tr>
<th>Form Title -- (appears above URL in most browsers and is used by WWW search)</th>
<th>Background Color:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q&amp;D Software Development Order Desk</td>
<td>FFFBF0</td>
</tr>
<tr>
<td>Form Heading -- (appears at top of Web page in bold type)</td>
<td>Text Color:</td>
</tr>
<tr>
<td>Q&amp;D Software Development Order Desk</td>
<td>000080</td>
</tr>
<tr>
<td>E-Mail responses to (will not appear on)</td>
<td>Alternate (for mailto forms only)</td>
</tr>
<tr>
<td><a href="mailto:dversch@q-d.com">dversch@q-d.com</a></td>
<td></td>
</tr>
<tr>
<td>Text to appear in Submit button</td>
<td>Text to appear in Reset button</td>
</tr>
<tr>
<td>Send Order</td>
<td>Clear Form</td>
</tr>
<tr>
<td>Scrolling Status Bar Message (max length = 200 characters)</td>
<td></td>
</tr>
<tr>
<td><em><strong>WebMania 1.5b with Image Map Wizard is here!!</strong></em></td>
<td></td>
</tr>
</tbody>
</table>

**Next Tab >>**
Simplicity, cont.

- **Good graphic design**
  - Few, well-chosen colors and fonts
  - Group with whitespace

- **Use concise language**
  - Choose labels carefully
How do users hold phones


To Do for Tuesday

- Make a password generator app
  - Asks user for: password length, # of numbers, # of symbols
  - Makes passwords following user specifications:
    - using a mix of upper & lower case letters
  - Display password
  - Button to generate new passwords (e.g., “refresh”)
  - Button to copy password to clipboard
    - Use Google to figure out how to copy things to clipboard