ECE 1778:
Creative Applications for Mobile Devices

Lecture 2

January 17, 2012

Today

1. Logistics/Organization of Course & Project
2. Introductions and Ideas, continued
   - Other half of class
   - Handout of Apper field description document
3. Introduction to Mobile Phone Programming Environment
   - Eclipse & Android Development Toolkit
   - Basic Concepts
   - Checkbox example
4. Overview of the Capabilities of Smartphones

Logistics

Some Logistics

- If you missed the first lecture see the on the course website:
  - http://www.eecg.utoronto.ca/~jayar/ece1778/
  - Look under Content
  - All lectures will be posted there

Sign up – sheets and ROSI

- If you did not, please sign up on the sign up sheets
- How many people here are not registered on ROSI?

- **Apper:** non-programmer, with expertise
  - from a specific discipline that app will leverage in significant way
- **Programmer**: capable of learning new environment fast
  - can be both, which means you can program well and have expertise in some specific field

Have You Started on the Assignment?

- Programmers: **P1**
  - Any issues/questions?
- Appers: **A1**
  - Any issues/questions?
- This is a lot of work to begin,
  - Necessary so you can do a project!

Assignments Due

A1 Part1: Was Due at 9am today

- P1 and A1 part 2 assignments due next week, 6pm, Monday January 23rd
- Submit via Portal – under Content.

Phones Available for Loan

- We have 18 Google Nexus S phones available for loan, for those who need them for assignments and the Project
  - Running Android 4.0

- Contact course TAs to borrow:
  - Daniel Di Matteo daniel.dimatteo@utoronto.ca
  - Braiden Brouseau braiden.brousseau@utoronto.ca
  - Day-long loans till ascertain demand

Many thanks to Google for the donation of these phones!

Recall: The Goal

- The goal of this course is to bring together people from different disciplines and to build an interesting/creative mobile application

- First Priority is to create those inter-disciplinary groups
  - We have more programmers than Appers,
  - Groups should 2 Programmers & 1 Apper

- Groups of 3 programmers will not be allowed

Stages of Project (aside from Assignments)

1. **Forming Groups**
   - Within the first 2 weeks
2. **One-Page Proposal**
   - Due January 31st; Must receive my approval to proceed
3. **Project Plan**
   - Due Feb 7th
4. **Proposal & Plan Presentations**
   - Weeks of February 14 and 28 [No class in Reading Week]
5. **Spiral 2 & Spiral 4 Presentations**
   - 2: March 6/13 4: March 20/27
6. **Final Presentations**
   - Weeks of April 3 & 10
7. **Final Report Due April 12th**

Extra Meeting to Form Groups

- Wednesday January 18th
- 6:30pm-7:30pm
- Sandford Fleming, room B560
  - After today's finishing introductions
  - Will find a way to help make matches there.

- Sandford Fleming is building south of Con Hall
- B560 is in basement, south side
  - In middle of Galbraith-Sandford Fleming buildings

Once You Have a Group

- Send email to:
  - Me (jayar@eecg.utoronto.ca)

- Provide:
  - Names, Department of each group member
  - Who is Programmer, Who is Apper (1 Apper, 2 Programmers preferred)
  - Student numbers
  - mobile platform you plan to do the project on
    - one of Android, iPhone (others require a special discussion)
    - if thinking about using Tablet
    - if you have your own device you can use

Note for iPhone/iPad Users

- Recall you must have a Mac to do this
- The University of Toronto has signed up under the University development program, see:
  - http://www.its.utoronto.ca/communication-and-collaboration/Apple_iOS_Developers_Centre.htm
  - Allows free download to device, which otherwise costs $US 99
  - Does not allow for app store distribution
  - (If you do pay $99 later, you will then be able to do app store)

Initial Thoughts/Pointers on Project

- You should be thinking of ideas for projects, as precursor to finding and forming your group
  - So you can have something to talk about on Wednesday

- Once you have a group:
  - If **Apper** in group, Apper needs to give rough idea of discipline
  - All groups: start kicking around ideas
  - Send me an email when you think you have something concrete that you can describe

- Create a Plan; be sure to use **Spiral/Agile** approach
  - Begin by making some small version work, and grow, incrementally from there

Introductions, continued

To Help in Project Group-forming

Introductions, Continued

- Last Day, half of the class introduced themselves
- Let’s do the other half
- Please take notes to keep track of people who you think might be compatible partners
- On Wednesday night, we’ll try to put people in some categories to help you explore matches.

- Don’t forget, the priority has to be on matching to Appers

Introduce Yourself, Round 2

1. Name
2. Taking Course for Credit – yes/no
3. What discipline you work in & **degree**
4. What your thesis topic is (if doing thesis)
5. If you work/worked, where & what you do/did.
6. Why you’re taking this course
7. What kind of phone you’re carrying
8. Apper: What idea you have for an app
9. Programmer: What you’re interested in doing app on.

Programmers:*
Mobile Phones and Android Development

*Some Should still be of interest to **Appers**

Mobile Phones are Very Small Computers

Good:

- The most portable computers ever
  - With built in sensors
- Amazing portals to the internet
- Can also make phone calls!

Not so good:

- Very small screens
- No/small keyboard
- Inexact pointing compared to mouse
- Processor speed and memory are slower/tighter than desktop
- Must make sure don’t interfere with a phone operation

Mobile Programming is *Event-Driven*

- Who is familiar with Event-Driven Programming?
  - Prevalent in graphical user-interfaces
- Different from straight-line procedural programming
  - Executed path is more linear – processing data in -> out
- Event-Driven
  - Flow of program determined by a series of user events
  - Sets up a series of user views
  - Waits to respond to events, such as:
    - User actions: button push, finger move, phone shake
    - System notifications – time elapsed, phone call, notification from internet
- Can be more complex because must handle different interacting patterns of events
  - shake + notification

An Android Application

- Is a series of windows (screens) presented to the user
  - Called ‘**Activities**’ in Android terminology

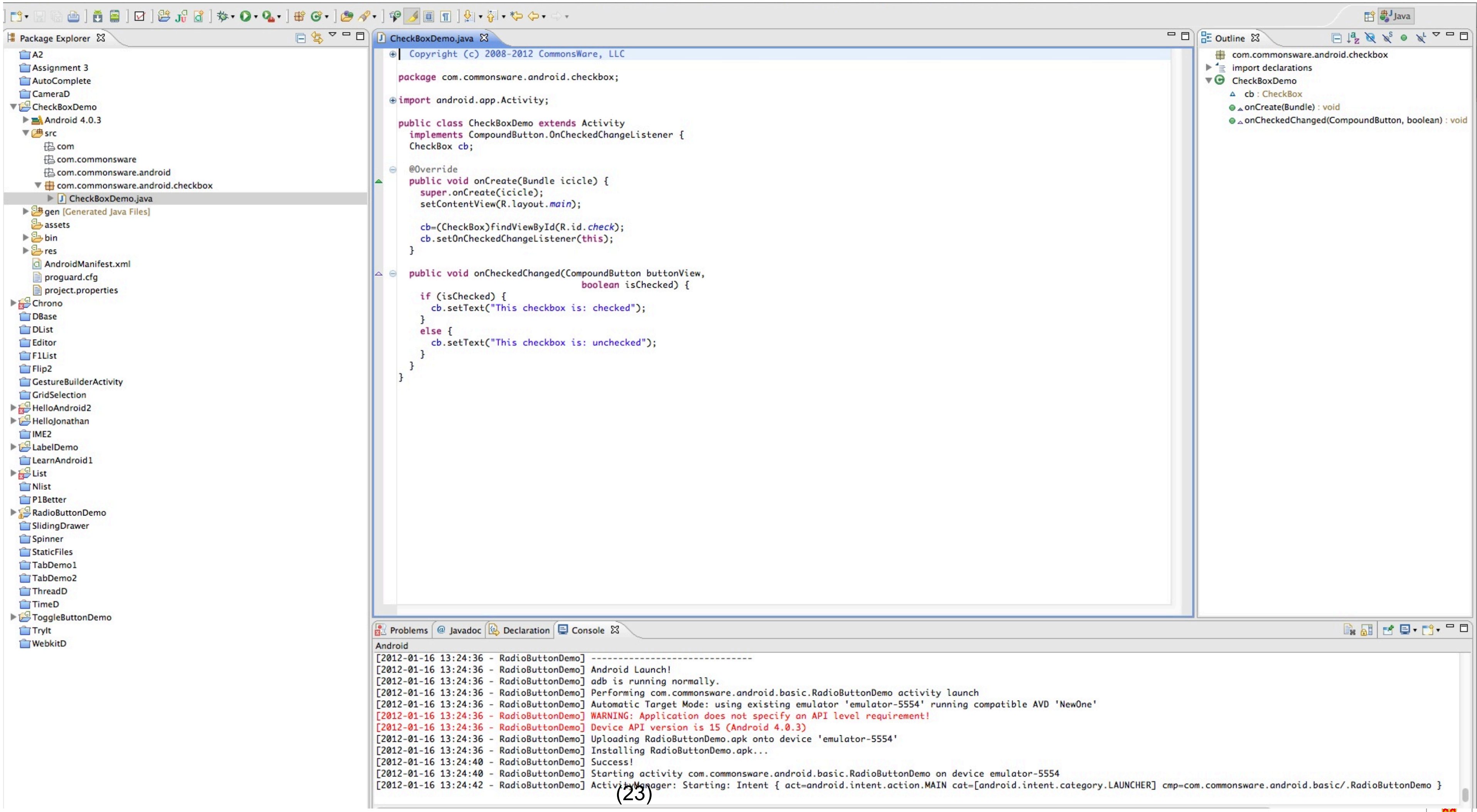
- Programmer ‘draws’ a rough picture of what each screen looks like
  - Each item in the screen is given an ID

- Source Code links itself to those IDs
  - Sets up listeners for touches
  - Allows changing of state

The Development Environment

- Eclipse
  - A GUI-based software development environment
  - Developed by IBM, open-sourced
- Android Software Development ToolKit (SDK)
  - Contains the libraries and tools needed to compile
  - Makes use of the Java compiler system from SUN
- Android Development Toolkit (ADT)
  - A plugin for Eclipse that allows use of the SDK from Eclipse

Eclipse Environment



```java
 Copyright (c) 2008-2012 CommonsWare, LLC

package com.commonsware.android.checkbox;

import android.app.Activity;

public class CheckBoxDemo extends Activity
  implements CompoundButton.OnCheckedChangeListener {
  CheckBox cb;

  @Override
  public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);

    cb=(CheckBox)findViewById(R.id.check);
    cb.setOnCheckedChangeListener(this);
  }

  public void onCheckedChanged(CompoundButton buttonView,
                                 boolean isChecked) {
    if (isChecked) {
      cb.setText("This checkbox is: checked");
    }
    else {
      cb.setText("This checkbox is: unchecked");
    }
  }
}
```

```
[2012-01-16 13:24:36 - RadioButtonDemo] ------------------------------
[2012-01-16 13:24:36 - RadioButtonDemo] Android Launch!
[2012-01-16 13:24:36 - RadioButtonDemo] adb is running normally.
[2012-01-16 13:24:36 - RadioButtonDemo] Performing com.commonsware.android.basic.RadioButtonDemo activity launch
[2012-01-16 13:24:36 - RadioButtonDemo] Automatic Target Mode: using existing emulator 'emulator-5554' running compatible AVD 'NewOne'
[2012-01-16 13:24:36 - RadioButtonDemo] WARNING: Application does not specify an API level requirement!
[2012-01-16 13:24:36 - RadioButtonDemo] Device API version is 15 (Android 4.0.3)
[2012-01-16 13:24:36 - RadioButtonDemo] Uploading RadioButtonDemo.apk onto device 'emulator-5554'
[2012-01-16 13:24:36 - RadioButtonDemo] Installing RadioButtonDemo.apk...
[2012-01-16 13:24:40 - RadioButtonDemo] Success!
[2012-01-16 13:24:40 - RadioButtonDemo] Starting activity com.commonsware.android.basic.RadioButtonDemo on device emulator-5554
[2012-01-16 13:24:42 - RadioButtonDemo] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.commonsware.android.basic/.RadioButtonDemo }
```

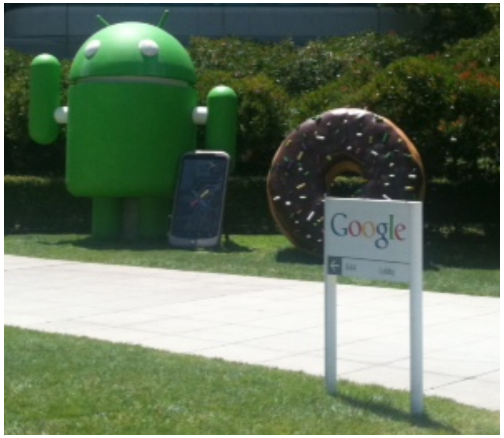
Projects and Targets

- To create an Android Application, must first create a **project**
  - Software directories that contain all of the files relating to the application

- Have to choose which version of Android to use …

Android Versions

- Google rapidly evolves Android:
  - 1.5 May 2009 = 3
  - 1.6 October 2009 = 4
  - 2.0/2.1 January 2010 = 5/6/7
  - 2.2 May 2010 = 8
  - 2.3 December 2010 = 9
  - 3.0 February 2011 = 11
  - 4.0 October 2011 = 15/16
- Each version has a name, in order: Cupcake, Donut, Éclair, Froyo and Gingerbread, Honeycomb, Ice Cream Sandwich



Project Structure – Android/Eclipse Folders

- **assets/**, static files you wish packaged with the application for deployment onto the device
- **bin/**, holds the compiled application
  - bin/classes/ compiled Java classes
  - bin/classes.dex executable created from compiled Java classes
  - bin/yourapp.ap_ holds your application's resources, packaged as a ZIP file (where yourapp is the name of your application)
  - bin/yourapp-*.apk is the actual Android application (where * varies)
- **gen/**, **generated** source code (by compiler)
- **libs/**, third-party Java JARs
- **src/**, your Java source code

Project Structure

- **res/**, "resources” - icons, GUI layouts
  - res/drawable/ for images (PNG, JPEG, etc.)
  - **res/layout/ for XML-based UI layout specifications**
  - res/menu/ for XML-based menu specifications
  - res/raw/ for general-purpose files
  - res/values/ for strings, dimensions, and the like
  - res/xml/ for other general-purpose XML files you wish to ship

Emulator vs. Real Phone

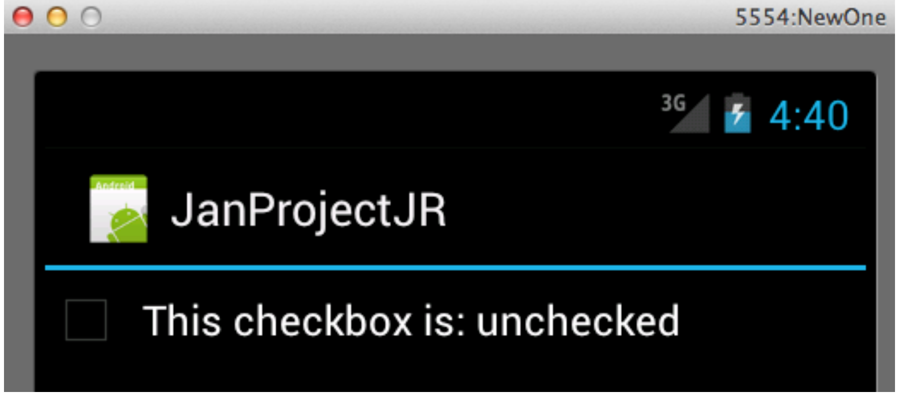
- You can run your application either an actual phone your want to run it on, or the emulator
  - The emulator is a software program running on the desktop that looks and acts like an Android phone
  - You can use it to test your basic programs/apps
  - The emulator camera actually works now!

Emulator

- Emulator is called an **‘Android Virtual Device’** or **AVD**
- There is some work in creating the device, as you have to specify various attributes of the fake phone, such as
  - Size of SD card memory
  - Which version of Android using
  - Size of screen
- Must properly set up phone to work
  - can have both up, SDK will ask which to use.

A Basic, Simple Program/App

- Goal – to make a simple check box:

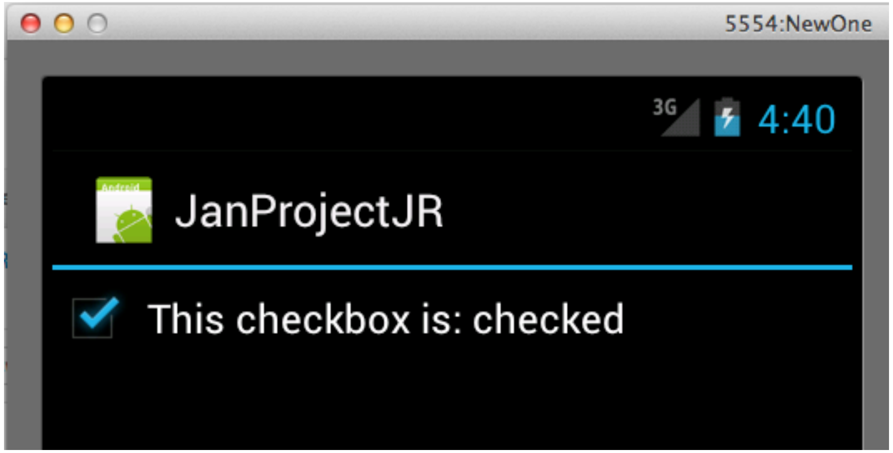


5554:NewOne

3G 4:40

JanProjectJR

□ This checkbox is: unchecked



5554:NewOne

3G 4:40

JanProjectJR

☑ This checkbox is: checked

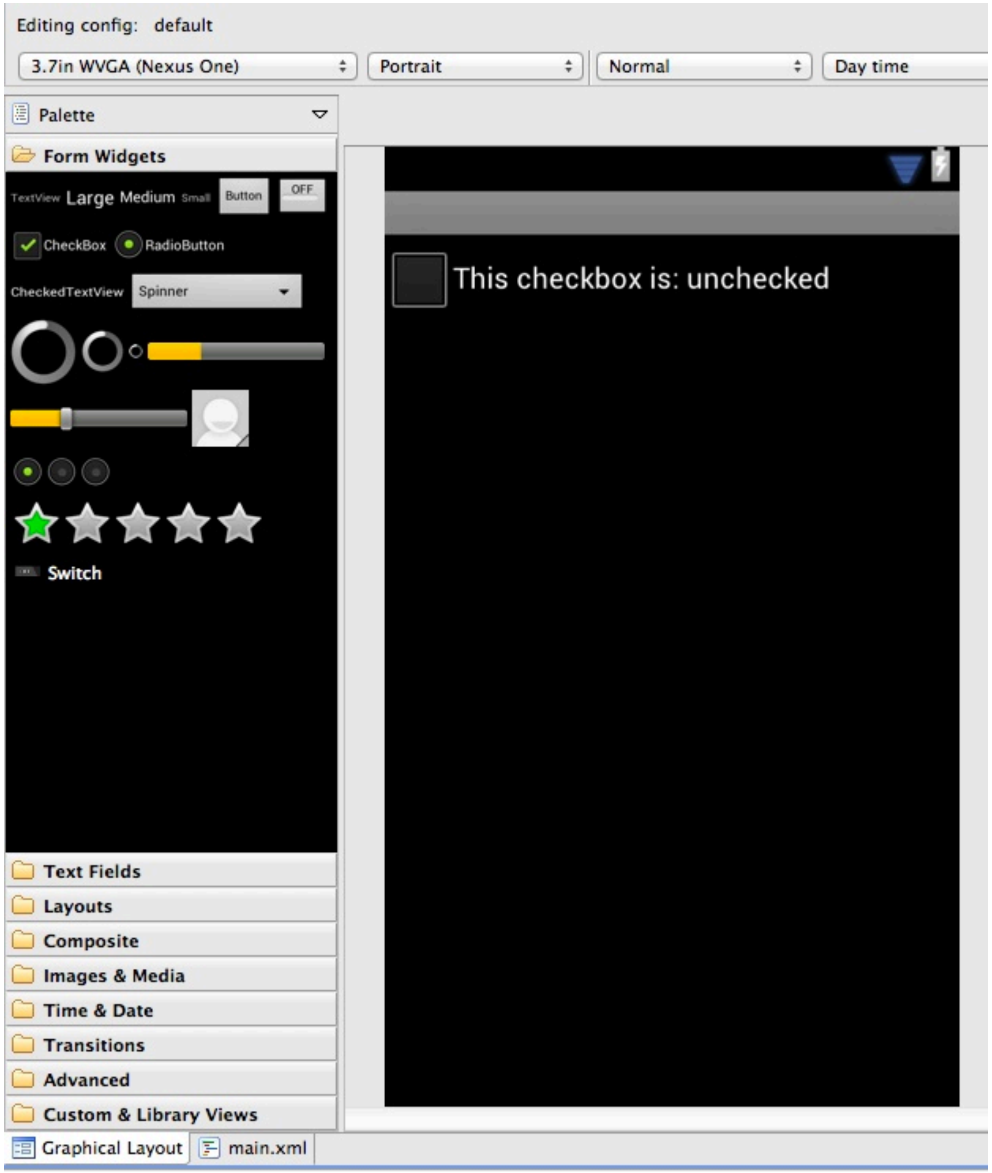
Describe the Layout in XML File

- Each activity has its own XML file that describes the different things on that screen.
- The following describes just 1 simple checkbox

```
<?xml version="1.0" encoding="utf-8"?>
<CheckBox xmlns:android="http://schemas.android.com/
apk/res/android"
    android:id="@+id/check"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This checkbox is: unchecked" />
```

Key – this name links into code

Can also use a GUI to Design Laout



Editing config: default

3.7in WVGA (Nexus One) | Portrait | Normal | Day time

This checkbox is: unchecked

This is just one ‘Widget’

- Checkbox is a widget
- If want more than one widget, need to place it within an organizing layout
  - e.g. ‘Linear Layout’

Code for CheckBox

```
/* a bunch of Java clases used:  */

Package eecg.utoronto.ca.checkbox;
import android.app.Activity;
import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.CompoundButton;
```

Code, cont'd

```
public class CheckBoxDemo extends Activity
  implements CompoundButton.OnCheckedChangeListener
{
CheckBox cb;
@Override
  public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);

    cb=(CheckBox)findViewById(R.id.check);
    cb.setOnCheckedChangeListener(this);
  }
```

Connects to 'main.xml'

Connects to @+id/check in main.xml

This 'listener is notified' when box changes

Routine called which box changed

```
public void onCheckedChanged(
          CompoundButton buttonView,
          boolean isChecked) {

  if (isChecked) { cb.setText ("This checkbox is: checked"); }

   else { cb.setText ("This checkbox is: unchecked");
    }
  }
}
```

Pointer to the check box changed

/* could have used **buttonView** instead of **cb** in this code */

Things to Demonstrate

- Eclipse Startup
- New Project
- Creating new Android Virtual Device (AVD)
- Running a project
- Placing a single widget
  - XML description
  - Switching between graphic view and XML in Eclipse
  - Properties
- Connection to Java Code through findViewById(R.id.*XXX);*

Other Widgets

- Button, ImageButton
  - Button to press, with special image
- Textview
  - Basic text label, changeable
- Imageview
  - Basic picture
- EditText
  - for entering text fields
- CheckBox
  - Ticking off an entry
- Radio Buttons

What Programmers Should Be Learning

- With Assignment 1:
  - After downloading the various elements of the programming environment
- Java basics if not already known
  - http://en.wikibooks.org/wiki/Java_Programming/Language_Fundamentals
  - Or some basic Java Text
  - I liked John Carter, '**Using Java**'
- Working within Eclipse
  - or, can choose to do everything in command/shell environment
  - lose some of Eclipse' good features
- Running the basic environment
- Understanding File Types in the Android Project

Overview of Smartphone Capabilities

To Get You Thinking about the Project

Based on iPhone, but Android Phones have same capabilities

A Smartphone is …

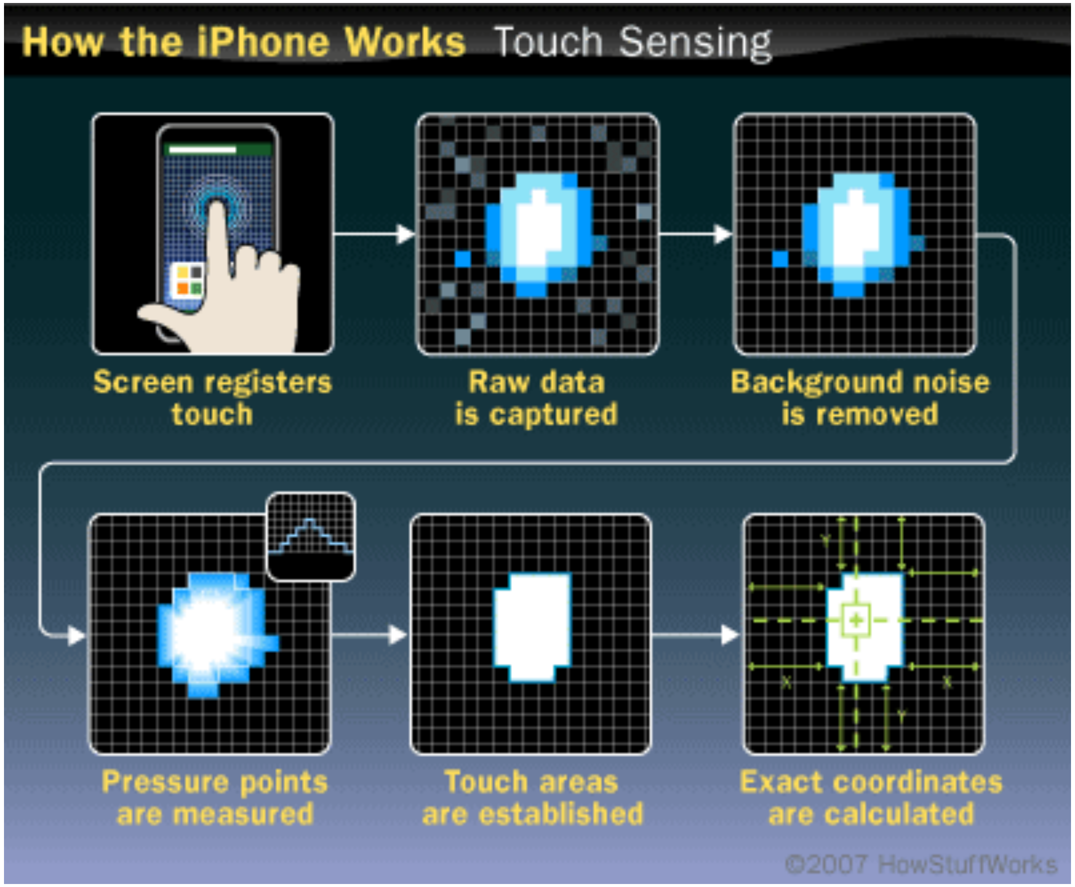
- A Computer small enough to unobtrusively carry, that
  - Is connected to the Internet – knowledge & compute power
  - Can **sense** its environment in many ways
  - Can **speak** to its environment in several ways
  - Can also make phone calls

- A *Computer*
  - Will do whatever you tell it to do, automating any drudgery
    - and never complain
  - Capable of sophisticated computation, including
    - analysis of its inputs
    - generating complex sound and images

Inputs and Sensors

Touch Screen

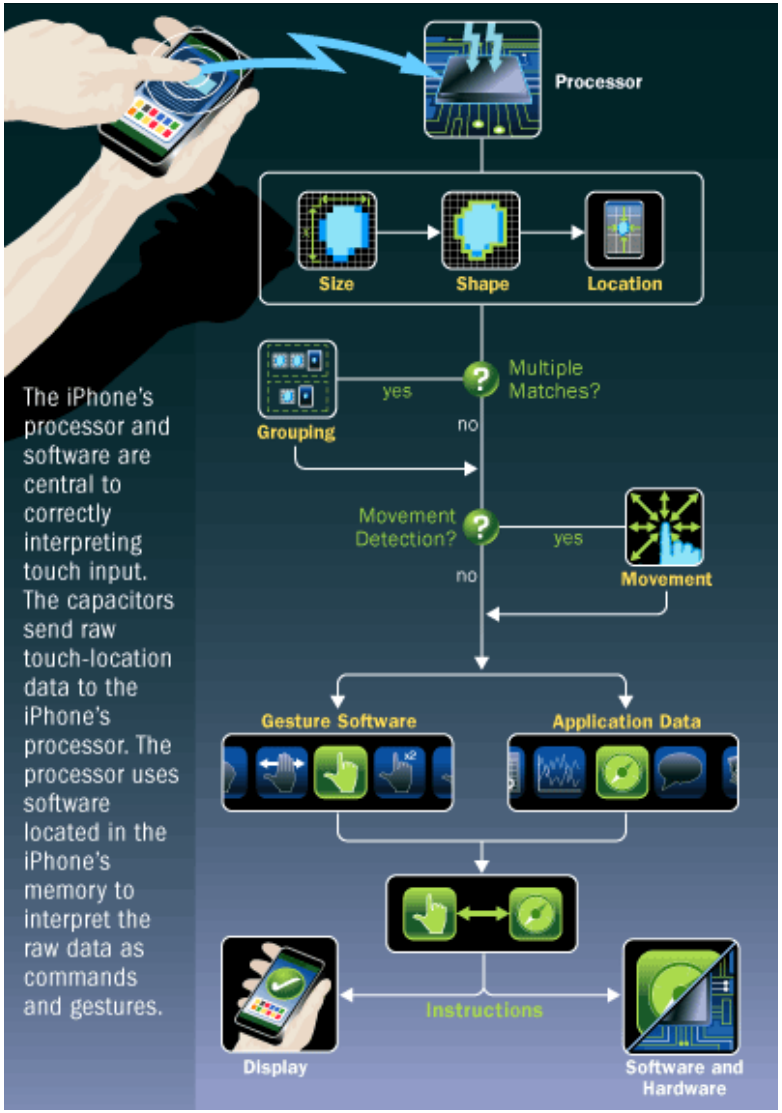
- The screen surface detects the touch of a finger
- Each touch can be turned into a specific coordinate


How the iPhone Works Touch Sensing
Screen registers touch
Raw data is captured
Background noise is removed
Pressure points are measured
Touch areas are established
Exact coordinates are calculated
Y
X
X
Y
©2007 HowStuffWorks

Touch Screen

- Coordinates can be turned into several different types of input:

1. Gestures
2. Selection actions
3. Tap counters
   - Double-tap
   - Triple-tap
4. Two Finger touch
5. Three Finger Touch …



The iPhone's processor and software are central to correctly interpreting touch input. The capacitors send raw touch-location data to the iPhone's processor. The processor uses software located in the iPhone's memory to interpret the raw data as commands and gestures.

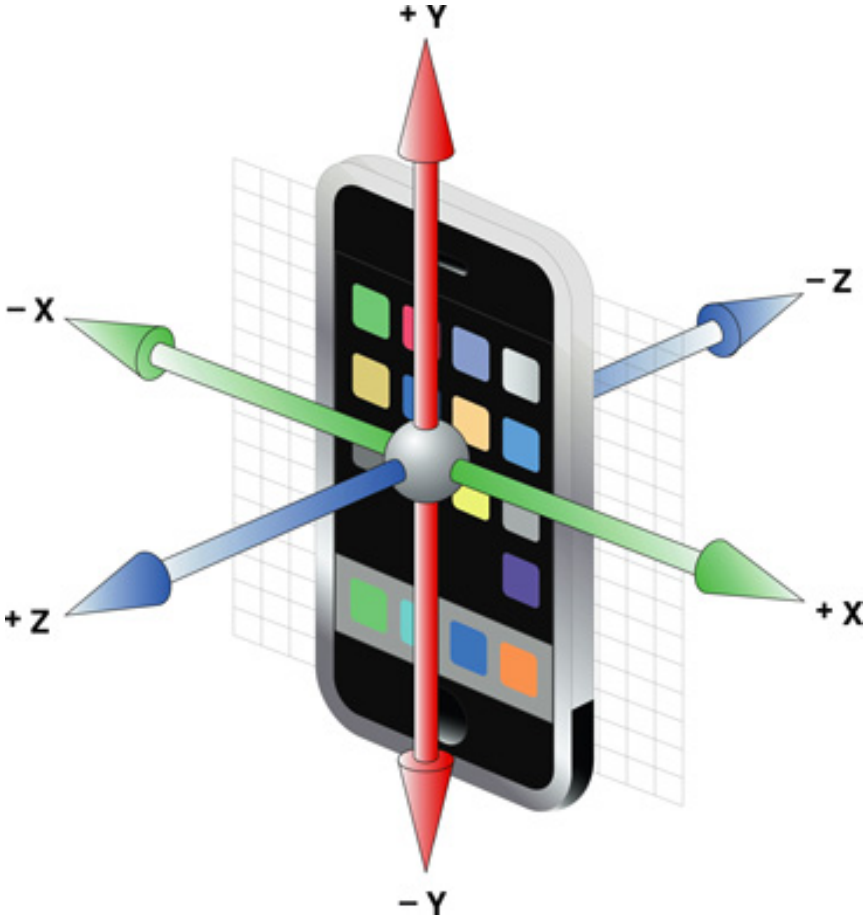
Can Touch Screen Be More?

- Could this sensor be used to measure something about the finger?
  - Blood flow
  - Blood Pressure
  - Heart Rate

- Use for?
  - Medical Diagnosis
  - Lie Detector

Accelerometer

- Can measure acceleration in 3-dimensions as shown
- Measured in $\mathbf{m/s^2}$
  - Remember your high school physics!
  - Get measurement in each dimension X,Y,Z
- Phone gives can give a ‘reading’ 100 times/s



Can Feel What the User is Doing

- Walking – step counting
- Running – speed measurement

- Can it tell something about the user’s Gait?
  - “Implementation of an iPhone as a wireless accelerometer for quantifying gait characteristics”
  - LeMoyne et. al, 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, August 31 - September 4, 2010
  - See other posts online

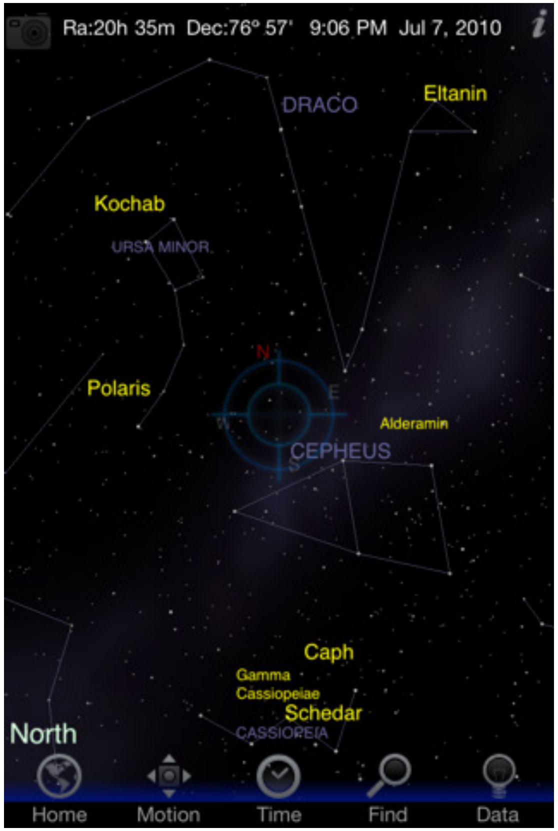
Other Motion Sensing

- Can tell if the phone is being shaken
  - Can use as an input
  - How sensitive is it?
  - Can it be used to measure Parkinsons tremors, in a medical application?

- Could perhaps detect if person fell down
  - could alert someone

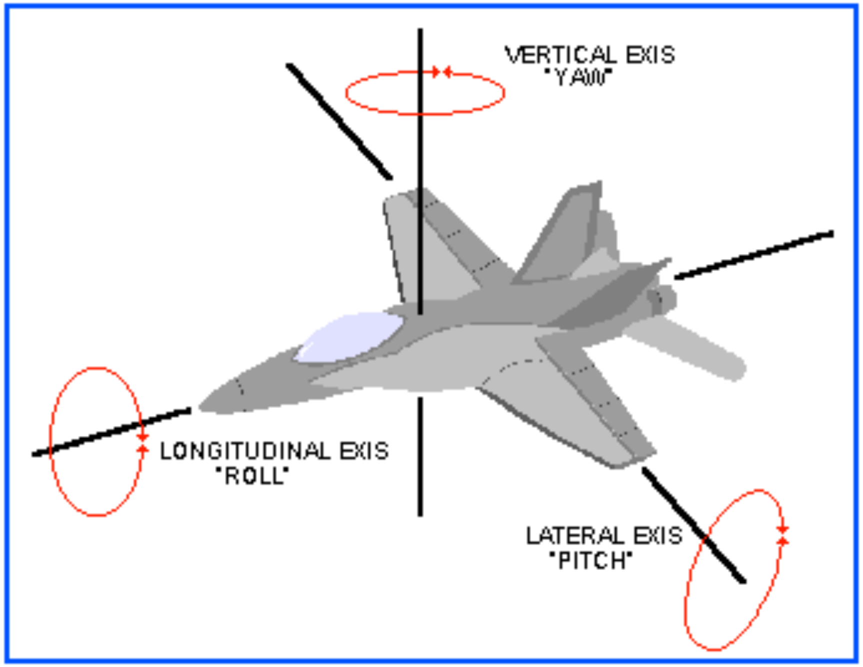
Motion Sensing with Accelerometer

- Gravity causes acceleration 9.8 m/s^2
  - If the phone is not accelerating (i.e. you're not moving it)
  - can determine the orientation of the phone,
  - by looking at which dimension has the '**G**':
    - X or Y or Z or some combination
- Used by stargazer apps to know where you're looking in the sky …



Gyroscope – measuring angular motion

- Gives: pitch, roll, and yaw
  - of phone, along X,Y,Z axis
- Rotation rate in radians/s
- Gives a better sense of the motion of the phone
- In iPhone 4 and Nexus S

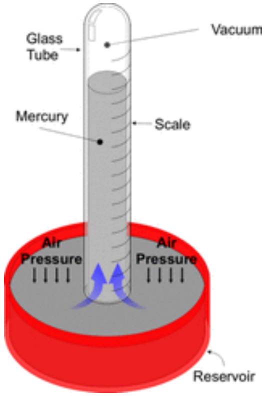


Compass

- Really a magnetometer
  - Can measure the magnetic field in 3 directions, X, Y, Z
  - Magnetic flux measured in micro-Tesla
  - Can use to make compass
  - Could also use as an instrument to measure presence of magnetic fields
- Where do magnetic fields exist?
  - Speakers, motors, screens, medical imaging, earth, big factories
- What are they used for?

Barometer

- Google Nexus S has Barometer
  - Measures atmospheric pressure
  - Change and rate of change gives an indication of weather
  - Measurement in hPA – hectoPascals
  - 1 atmosphere = 1013 hPA



- Could use as altimeter
- What could crowd-sourced pressure measurements reveal?

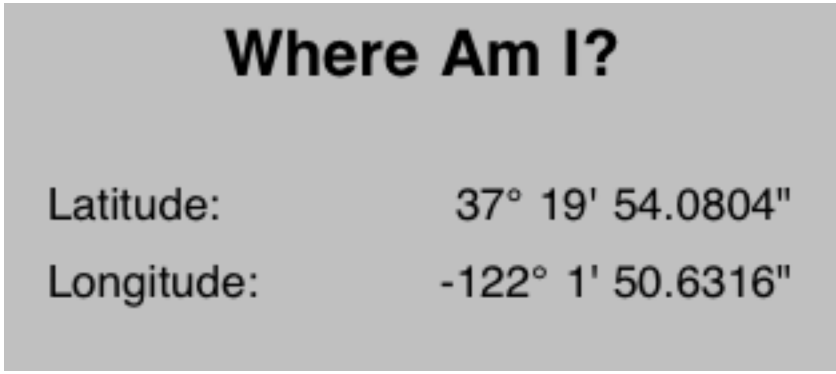
Weather Prediction Using Barometer

- Decreasing barometric pressure indicates storms, rain and windy weather.
- Rising barometric pressure indicates good, dry, and colder weather.
- Slow, regular and moderate falls in pressure suggest a low pressure area is passing in a nearby region. Marked changes in the weather where you are located are unlikely.
- Small rapid decreases in pressure indicate a nearby change in weather. They are usually followed by brief spells of wind and showers.
- A quick drop in pressure over a short time indicates a storm is likely in 5 to 6 hours.
- Large, slow and sustained decreasing pressure forecasts a long period of poor weather. The weather will be more pronounced if the pressure started rising before it began to drop.
- A rapid rise in pressure, during fair weather and average, or above average pressure, indicates a low pressure cell is approaching. The pressure will soon decrease forecasting poorer weather.
- Quickly rising pressure, when the pressure is low, indicates a short period of fair weather is likely.
- A large, slow and sustained rise in pressure forecasts a long period of good weather is on its way.

Global Positioning Satellite Receiver

GPS

- Can determine the location of the phone in the geographic coordinate system
- Quickly accurate to within 100 metres, takes longer to do better
  - Does not work inside buildings
  - Will have more trouble when lots of buildings around
- Knowing where you are is incredibly useful in business; can provide context for assistive apps



Where Am I?

Latitude:	37° 19' 54.0804"
Longitude:	-122° 1' 50.6316"

Location Services

- In Android, there are several ‘Location Services’ available that include the GPS
  - GPS uses a fair bit of power, can’t have on all the time without draining the battery quickly

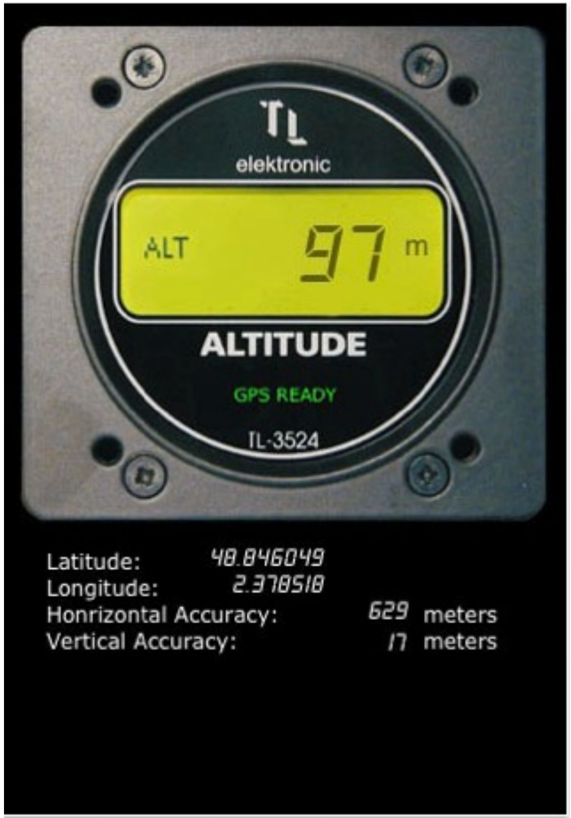
- The other two methods of locating are using:
  1. Cell phone tower triangulation
  2. Wifi Network IP addresses of the routers

GPS

- Can also get the raw position of the satellites themselves

Altimeter

- Using the GPS, can also determine the height of the phone



Latitude: 48.846049
Longitude: 2.378518
Honrizontal Accuracy: 629 meters
Vertical Accuracy: 17 meters

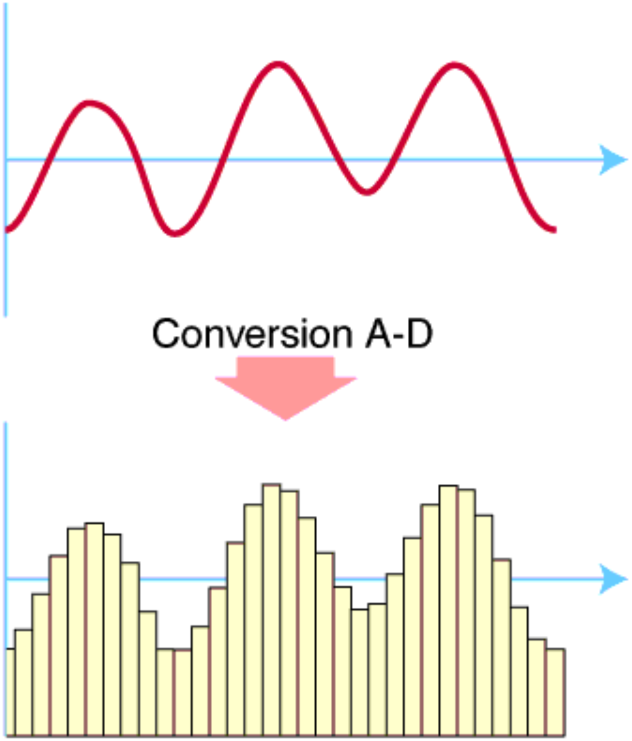
(57)

Ambient Light Sensor

- Used for measuring ambient light to set screen brightness
- Measures the light, in Lux, coarsely
  - Seem like roughly 8 different values, on the Nexus One

Microphone

- Converts sound into data
  - Microphone converts sound waves into voltage
    - Which varies over time
  - Circuit converts voltage into into digital values
  - Sound becomes a series of digital values
    - Get samples at 48K samples/s
    - Good quality sound!
- Sound Processing
  - aka **Digital Signal Processing**



Conversion A-D

Sound Processing Example 1

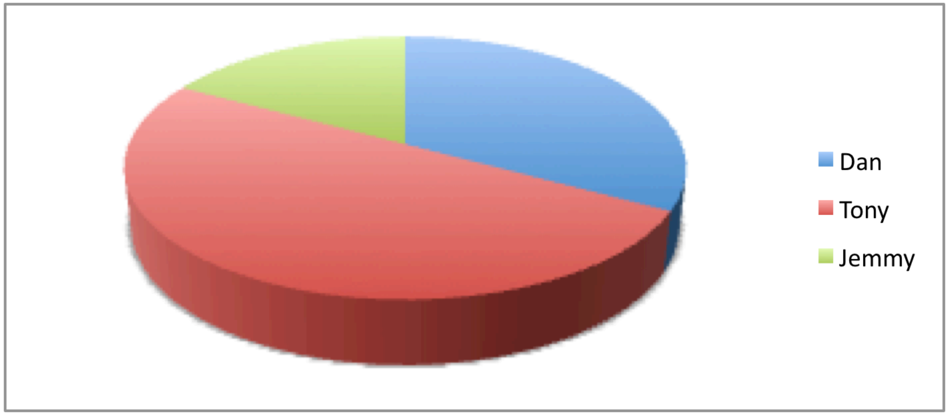
- Famous **Shazam** app
  - Listens to 15 seconds of song playing
  - Can tell you what the song is
  - Sends sound sample up to server to do this work
  - Lets you buy song

- Most processing is done on a server

Sound Processing Example 2

- Listen to a conversation, and measure the fraction of the conversation that each participant takes up!

- Daniel DiMatteo's, 4th Year Undergraduate
  - Known as 'Diarization'

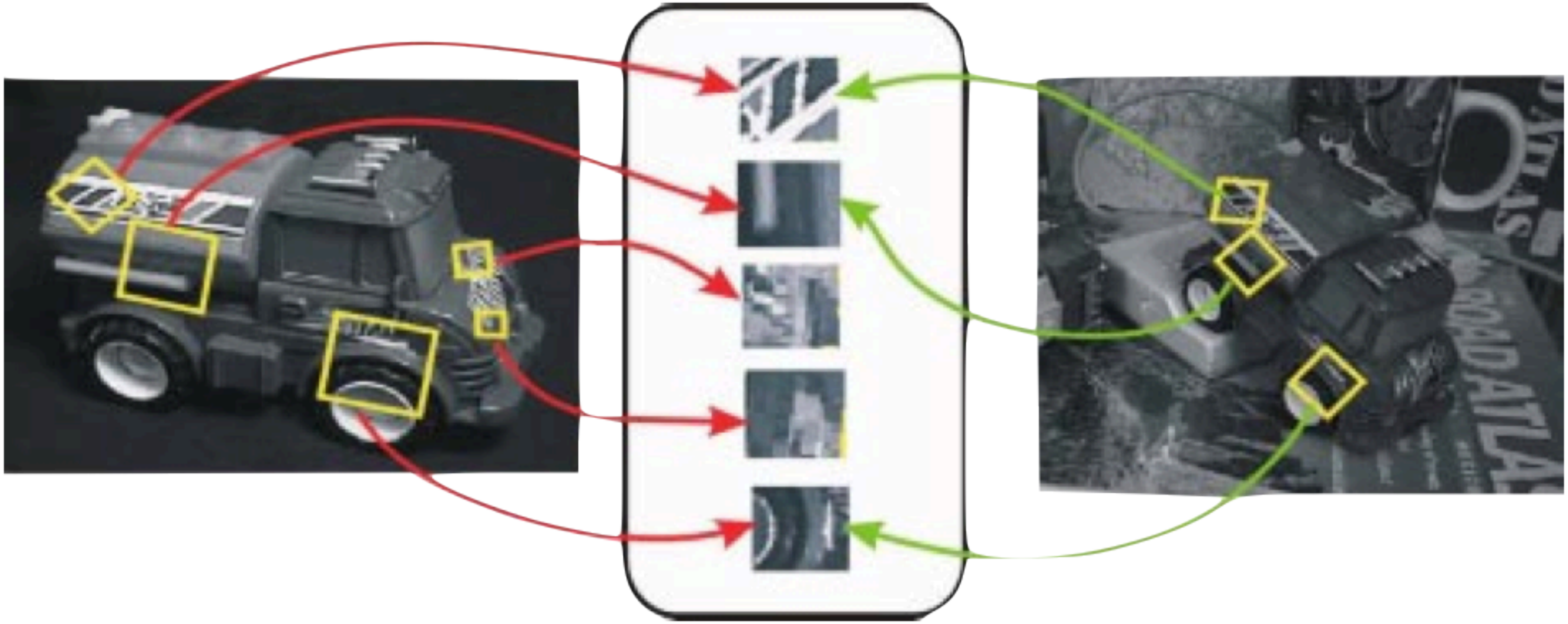


Back Camera

- Can record images
  - Large files with high resolution
    - 2MPixels – 8 MPixels
- Can record video
  - ~ 30 frames/second of pictures
- Can we use it to “see things”?
  - Yes!
  - Computer Vision field
  - Difficult, slow

Computer Vision

- Computation to convert *many* pixels to information
- Computers ‘see’ in much the same way that people do



Computer Vision

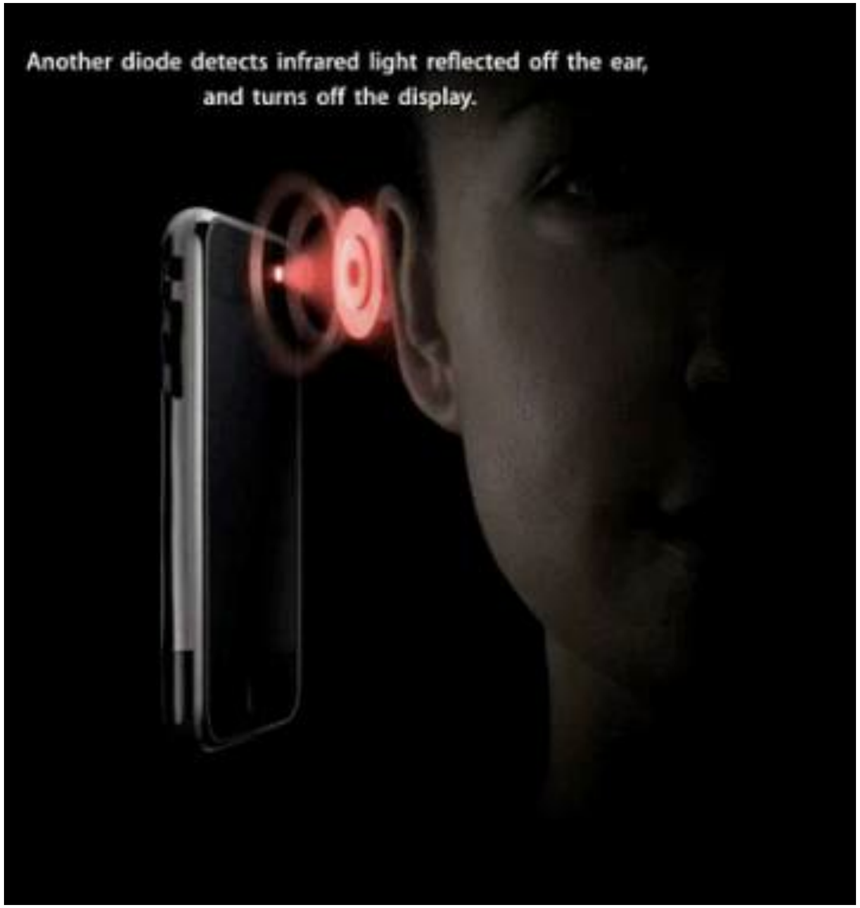
- Often too slow to do in real time, but not always
- There is some open-source software, **OpenCV**, which can do many things, but not very quickly
  - Has been ported to Android
- Braiden Brousseau's (TA) Master's thesis is about speeding up OpenCV on Android using an FPGA
- He can help with using OpenCV

Front Facing Camera

- Allows for video interaction
  - Skype uses this
  - Lower resolution than back camera
- Can look at you and see how you're feeling
- Can maybe track your eye movements as you watch things
- Diagnose depression?
  - eyes are the window to the soul

Proximity Sensor

- Can detect if phone is near to something, particularly the head
- Used to turn off touch screen when phone is to near to ear
- Simple Near/Not input
  - Doesn’t give distance, yet


Another diode detects infrared light reflected off the ear, and turns off the display.

Humidity and Temperature Sensors

- Apparently, some Android devices have a humidity sensor, but it is for sensing if you’ve dropped the device in water, and just turns colour and is permanently triggered

- There is also a temperature sensor, but it is just for the battery, not the ambient temperature (yet)

Output Devices

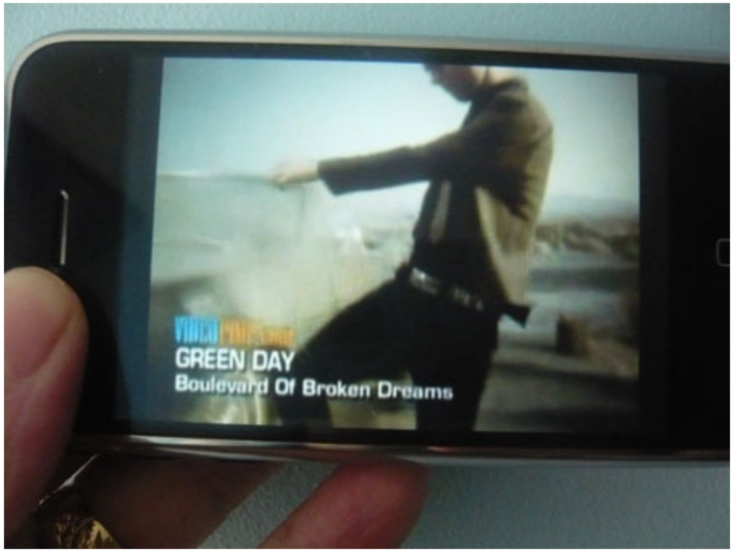
Hi-Resolution Screen

- Most recent phones have very high quality screens
  - Quality is the # pixels
- Resolution of Samsung Google Nexus S
  - 480x800 total resolution
- Cheaper phones have less:
  - Hua Wei U8100 240x320
  - Alcatel OT-981A 240x320

Video

- Special hardware to enable 30 frames/second video

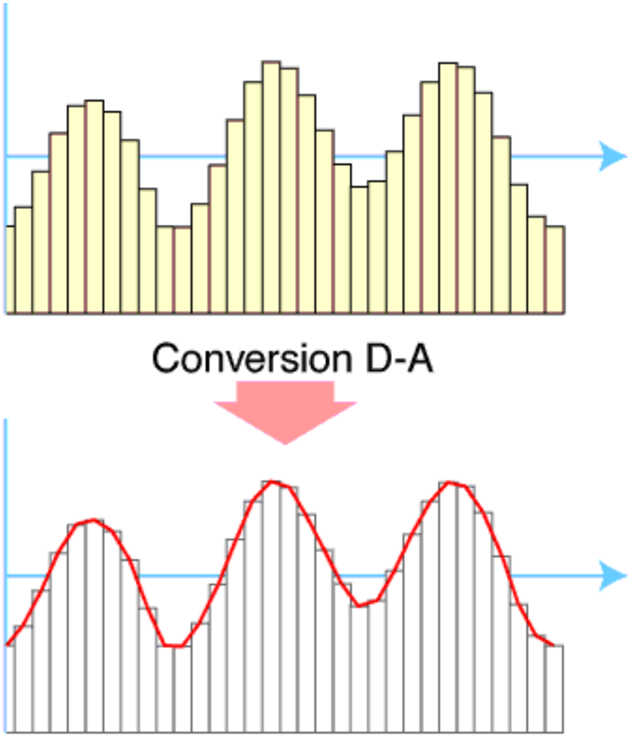
- Displaying video can use up much or all of the processor’s computational capacity;
  - Most phones have special hardware to handle this task
  - Nexus S has MP4/H.264/H.263 player



Speakers/Audio Out

- Sound Output
  - Two speakers
    - Quiet one for ear
    - Loud speaker

- Play previously recorded files
  - Should be able to do **text-to-speech**
- Many possible sound filters –
  - Auto-tune voices to make at right pitch
  - Make funny voices
  - Synthesized Musical instruments



Vibration Output

- Can create a short buzz
- Can control vibration pattern, duration and intensity
- This can be a significant output device – ‘haptic’ feedback

Camera Flash

- Bright White LED
  - Meant for taking pictures
  - Can be used to light up a room
  - Signal someone
  - (transmit data?)

- Undergraduate project:
  - Evoke red-eye effect **on purpose**
  - Is a picture of retina
  - To do eye-disease diagnosis
    - with computer vision

The Computer: Storage, Networking and External Devices

Computer

- What can a computer do?
  - Processors are powerful
  - Nexus S has 1GHz ARM Cortex A8 processor

- Many things!
  - Optimization
  - Search
  - Sort
  - Artificial Intelligence

Storage Capacity

- Local storage of 2 to 64 Gbytes of permanent storage
  - Flash-based solid-state disk

- Can load many databases locally onto the device
  - Dictionaries, no problem!
  - Maps
  - Phonebooks
  - Location Services

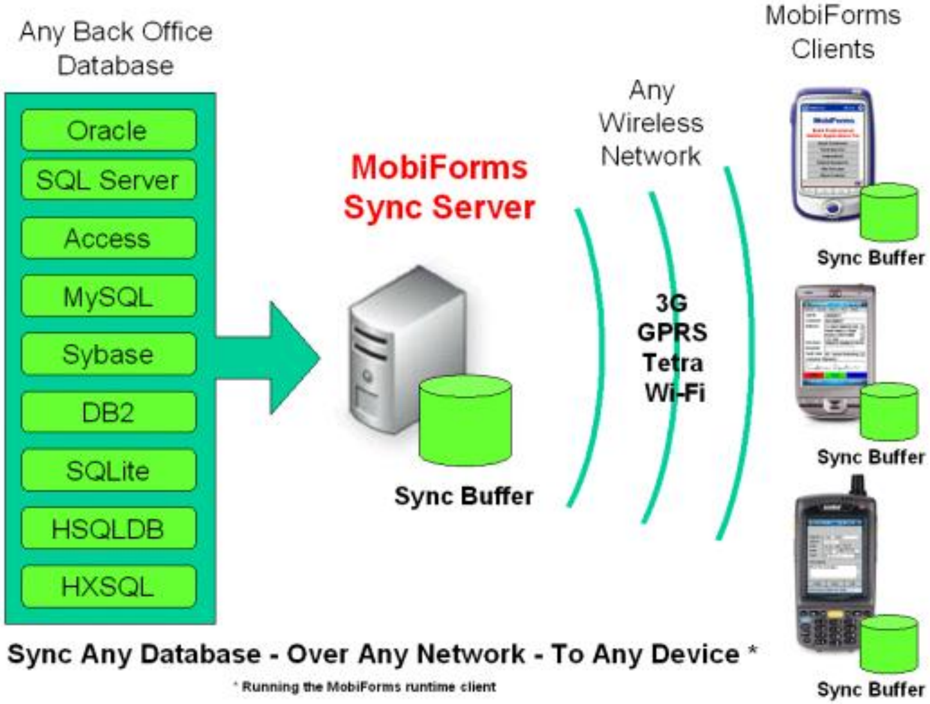
Network – 3G/Wifi: Gateway to the Internet

- All phones have at least 2 ways to talk to the internet
  - Local WIFI
  - 3G cellular data networks
- Connection to more computing and storage
- Connection to other phones

Map of the Internet, The Opte Project, www.opte.org

Not Just App: Probably Need Web Site

- Many apps need 'backing' website/ database
- Provides phone with:
  - Communication to other people
  - Data
  - Backup
  - Information from Internet



Bluetooth Connection

- Connect to a whole class of external devices, wirelessly
  - earphones
  - small spy cameras

- Could be important way to add other devices without physical connection
  - Make use of phone's capabilities without holding it

Using All These & More

- Come up with something interesting in your field
- Make it work!

Tomorrow: Meeting to Form Groups

- Wednesday January 18th (Tomorrow)
- 6:30pm-7:30pm
- Sandford Fleming, room B560
  - Will do super-fast introductions – name, field, interest

- Sandford Fleming is building south of Con Hall
- B560 is in basement, south side
  - In middle of Galbraith-Sandford Fleming buildings