About Me
A Little about me

- WPI Computer Science Professor
- Research interests:
  - mobile computing especially mobile health, computer graphics
- Started working in mobile computing, wireless in grad school
- CS + ECE background (Hardware + software)
- Current active research: Mobile health apps
  - E.g: AlcoGait app to detect how drunk Smartphone owner is
    - [https://www.youtube.com/watch?v=pwZaoKmfq8c](https://www.youtube.com/watch?v=pwZaoKmfq8c)
Administrivia
Administrivia: Schedule

- **Week 1-7**: I will introduce class, concepts, Android (Students: Android programming, machine learning, 4 assigned projects)
  - **Goal**: Students acquire basic Android programming skills to do excellent project
- **Week 8-12**: Students will present 1 paper each in groups
- **Week 9**: Students will present final project proposal
- **Week 9-14**: Students work on final project
- **Week 14**: Students present + submit final projects
- Quizzes (5) throughout
Requirements to get a Grade

- **Projects**: 4 assigned (40%) and 1 final project(s) (25%)
- **Final project phases**: (See class website for deadlines)
  1. Pick partners, form project groups
  2. Submit 1-slide of proposed idea (problem + envisioned solution)
  3. Present project proposal
     + plus submit proposal (intro + related work + methodology/design + proposed project plan)
  4. Build app, evaluate, experiment, analyze results
  5. Present results + submit final paper (in week 7)

- **Grading policy**: Presentation 15%, Assigned Projects 40%, Final project: 25%, Quizzes: 20%
Course Texts

- **Android Texts:**
  - *Head First Android Development*, Dawn and David Griffiths, O'Reilly, 2015

- Will also use official Google Android documentation
- Learn from research papers: Why not text?
Class in 2 Halves

- 2 Halves: About 1 hr 15 mins
- Break of about 20 mins
- Come and meet me at the end not during break
  - I need break too
Poll Question

- How many students:
  1. **Own** recent Android phones (running Android 4.4, 5, 6, 7 or 8?)
  2. **Can borrow** Android phones for projects (e.g. from friend/spouse)?
  3. **Do not own and cannot borrow** Android phones for projects?
Mobile Devices
Mobile Devices

- Smart phones (Blackberry, iPhone, Android, etc)
- Tablets (iPad, etc)
- Laptops
SmartPhone Hardware

- **Smart = Communication + Computing + Sensors**
  - **Communication:** Talk, text, Internet access, chat
  - **Computing:** Java apps, JVM, apps
    - Powerful processors: Quad core CPUs, GPUs
  - **Sensors:** Camera, video, accelerometer, heart rate sensor, etc

- Google Pixel XL phone: Quad core 1.6 GHz Snapdragon CPU, Adreno 530 GPU, 4GB RAM
  - A PC in your pocket!!

<table>
<thead>
<tr>
<th></th>
<th>Nexus 4</th>
<th>Galaxy S III</th>
<th>iPhone 5</th>
<th>Moto Droid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>APQ8064</td>
<td>MSM8960</td>
<td>Apple A6</td>
<td>OMAP 3430</td>
</tr>
<tr>
<td></td>
<td>1.7 GHz</td>
<td>1.7 GHz</td>
<td>1.3 GHz</td>
<td>600 MHz</td>
</tr>
<tr>
<td></td>
<td><strong>Quad-core</strong></td>
<td><strong>Dual-core</strong></td>
<td><strong>Dual-core</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>Adreno 320</td>
<td>Adreno 225</td>
<td>PowerVR SGX543MP3</td>
<td>PowerVR SGX 530</td>
</tr>
<tr>
<td></td>
<td>OpenGL ES 3.0</td>
<td>OpenGL ES 2.0</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
</tr>
<tr>
<td></td>
<td>OpenCL 1.2</td>
<td>OpenVG 1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NA</td>
<td>400 MHz</td>
<td>266 MHz (<strong>Tri-core</strong>)</td>
<td>200 MHz (1.6 GFLOPS)</td>
</tr>
<tr>
<td></td>
<td><strong>40-45 GFLOPS</strong></td>
<td><strong>19.2 GFLOPS</strong></td>
<td><strong>25.5 GFLOPS</strong></td>
<td></td>
</tr>
</tbody>
</table>

**GLOPS:** floating-point operations per second

Comparison courtesy of Qian He (Steve)
Smartphone Sensors

- Typical smartphone sensors today
  - accelerometer, compass, GPS, microphone, camera, proximity
- Can be used for intelligent sensing/adaptive applications
Growth of Smartphone Sensors

- Every generation of smartphone has more and more sensors!!

Future sensors?
- Complex activity sensor,
- Pollution sensor,
- etc

Image Credit: Qualcomm
Wireless Networks
Wireless Network Types

- **Wi-Fi (802.11)**: (e.g. Starbucks Wi-Fi)
- **Cellular networks**: (e.g. Sprint network)
- **Bluetooth**: (e.g. car headset)
- **Near Field Communications (NFC)**
  - e.g. Mobile pay: swipe phone at dunkin donut
## Wireless Networks Comparison

<table>
<thead>
<tr>
<th>Network Type</th>
<th>Speed</th>
<th>Range</th>
<th>Power</th>
<th>Common Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLAN</td>
<td>600 Mbps</td>
<td>45 m – 90 m</td>
<td>100 mW</td>
<td>Internet.</td>
</tr>
<tr>
<td>LTE (4G)</td>
<td>5-12 Mbps</td>
<td>35km</td>
<td>120 – 300 mW</td>
<td>Mobile Internet</td>
</tr>
<tr>
<td>3G</td>
<td>2 Mbps</td>
<td>35km</td>
<td>3 mW</td>
<td>Mobile Internet</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>1 – 3 Mbps</td>
<td>100 m</td>
<td>1 W</td>
<td>Headsets, audio streaming.</td>
</tr>
<tr>
<td>Bluetooth LE</td>
<td>1 Mbps</td>
<td>100+ m</td>
<td>.01–.5 W</td>
<td>Wearables, fitness.</td>
</tr>
<tr>
<td>NFC</td>
<td>400 kbps</td>
<td>20 cm</td>
<td>200 mW</td>
<td>Mobile Payments</td>
</tr>
</tbody>
</table>

Table credit: Nirjoin, UNC

Different speed, range, power, uses, etc
Mobile Computing
mobile

adjective
/ˈməbəl, ˈmō.bīl/

1. able to move or be moved freely or easily.
   "he has a major weight problem and is not very mobile"
   synonyms: able to move (around), moving, walking; motile; ambulant
Mobile Computing

- Human computes while moving
  - Continuous network connectivity,
  - Points of connection (e.g. cell towers, WiFi access point) might change
- **Note:** Human initiates all activity, (e.g launches apps)
- Wireless Network is *passive*
- **Example:** Using *foursquare.com* on SmartPhone
Mobile computing = computing while location changes

**Location-aware:** Location must be one of app/program’s inputs

Different user location = different output (e.g. maps)

**E.g.** User in California gets different map from user in Boston
Location-Aware Example

- Location-aware app must have different behavior/output for different locations
- Example: Mobile Yelp

  - **Example search**: Find Indian restaurant

  - App checks user’s location

  - Indian restaurants **close to user’s location** are returned
Example of Truly Mobile App: Word Lens

- Translates signs in foreign Language

- Location-dependent because sign location varies
Some Mobile apps are not Location-Aware

- If output does not change as location changes, not location-aware
- Apps run on mobile phone just for convenience
- Examples:

  - Diet recording app
  - Mobile banking app
Which of these apps are Location-Aware?

a. Yahoo mail mobile

b. Uber app
Mobile Device Issue: Energy Efficiency

- Most resources increasing exponentially except battery energy (ref. Starner, IEEE Pervasive Computing, Dec 2003)

Some Strategies:

- **Energy harvesting**: Energy from vibrations, charging mats, moving humans
- **Scale content**: Reduce image, video resolutions to save energy
- **Better user interface**: Estimate and inform user how long each potential task will take
  - E.g: At current battery level, you can either type your paper for 45 mins, watch video for 20 mins, etc

![Graph showing improvements in laptop technology from 1990-2001.](image)
Ubiquitous Computing
ubiquitous
/yooˈbikwətəs/

adjective

present, appearing, or found everywhere.
"his ubiquitous influence was felt by all the family"
synonyms: omnipresent, ever-present, everywhere, all over the place, pervasive,
Ubiquitous Computing

- Collection of specialized assistants to assist human in tasks (reminders, personal assistant, staying healthy, school, etc)
- App figures out user’s current state, intent, assists them
- **How?** array of *active* elements, sensors, software, Artificial intelligence
- Extends *mobile computing* and *distributed systems* (more later)
- **Note:** System/app initiates activities, has intelligence
- **Example:** Google Now app, updates user on
  - Driving time to work, home
  - Weather
  - Favorite sports team scores, etc
Ubicomp Senses User’s Context

- Context?
  - **Human:** motion, mood, identity, gesture
  - **Environment:** temperature, sound, humidity, location
  - **Computing Resources:** Hard disk space, memory, bandwidth
  - **Ubicomp example:**
    - **Assistant senses:** Temperature outside is 10F (environment sensing) + Human plans to go work (schedule)
    - **Ubicomp assistant advises:** Dress warm!

- Sensed **environment + Human + Computer resources = Context**

- **Context-Aware** applications adapt their behavior to context
Sensing the Human

- Environmental sensing is relatively straight-forward
  - Use specialized sensors for temperature, humidity, pressure, etc

- Human sensing is a little harder (ranked easy to hard)
  - **When**: time (Easiest)
  - **Where**: location
  - **Who**: Identification
  - **How**: (Mood) happy, sad, bored (gesture recognition)
  - **What**: eating, cooking (meta task)
  - **Why**: reason for actions (extremely hard!)

- Human sensing (gesture, mood, etc) easiest using cameras

- Research in ubiquitous computing integrates
  - location sensing, user identification, emotion sensing, gesture recognition, activity sensing, user intent
**Sensor**

- **Example:** E.g. door senses only human motion, opens
- **Sensor:** device that can sense physical world, programmable, multi-functional for various tasks (intrusion detection, temperature, humidity, pressure, etc)
- More generally means device that can take input from physical world
  - Also includes camera, microphone, etc
- Ubicomp uses data from sensors in phone, wearables (e.g. clothes), appliances, etc.

(courtesy of MANTIS project, U. of Colorado)  
RFID tags  
Tiny Mote Sensor, UC Berkeley
Ubiquitous Computing: Wearables
Ubiquitous Computing: Wearable sensors for Health

remote patient monitoring

UbiComp: Wearables, BlueTooth Devices

Body Worn Activity Trackers

External sources of data for smartphone
Definitions: Portable, mobile & ubiquitous computing
Distributed Computing

- Computer system is physically distributed
- User can access system/network from various points.
- E.g. Unix cluster, WWW
- Huge 70’s revolution

**Distributed computing example:**
- WPI students have a CCC account
- Log into CCC machines,
- Web surfing from different terminals on campus (library, dorm room, zoolab, etc).

**Finer points:** network is fixed, Human moves
Portable (Nomadic) Computing

**Basic idea:**
- Network is fixed
- Device moves and changes point of attachment
- No computing while moving

**Portable (nomadic) computing example:**
- Mary owns a laptop
- Plugs into her home network,
- **At home:** surfs web while watching TV.
- Every morning, brings laptop to school, plug into WPI network, boot up!
- **No computing while traveling to school**
Mobile Computing Example

- Continuous computing/network access while moving, automatic reconnection

**Mobile computing example:**
- John has SPRINT PCS phone with web access, voice, SMS messaging.
- He runs apps like facebook and foursquare, continuously connected while walking around Boston

**Finer points:**
- John and mobile users move
- Network deals with changing node location, disconnection/reconnection to different cell towers
Ubiquitous Computing Example

- **Ubiquitous computing:** John is leaving home to go and meet his friends. While passing the fridge, the fridge sends a message to his shoe that milk is almost finished. When John is passing grocery store, shoe sends message to glasses which displays “BUY milk” message. John buys milk, goes home.

- **Core idea:** ubiquitous computing assistants **actively** help John
SmartPhone Sensing
Smartphone Sensing

- Smartphone used to sense human, environment

  *Example:* Human activity sensing (e.g. walking, driving, climbing stairs, sitting, lying down)

- *Example 2:* Waze crowdsourced traffic
Sensor Processing

- **Machine learning** commonly used to process sensor data
  - Action to be inferred is hand-labelled to generate training data
  - Actual data is mined for combinations of sensor readings corresponding to action
What Can We Detect/Infer using These Sensors

Smartphone Sensing!!

Smartphone Sensor data

Machine Learning

Eating/Drinking

Stress, Mood

Activity

Social interactions

Mobility patterns

Cardiac health

Sleep Cycle

Conversations

Sleep Quality

Image Credit: Deepak Ganesan, UMass
Internet of Things (IoT)
IoT: Networked Smart Things (Devices)

- Smart things: Can be accessed, controlled over the network, learns users patterns

Nest Smart thermostat
- Learns owners manual settings
- Turns down heat when not around

Smart Fridge
- See groceries in fridge from anywhere
Other Ubicomp Systems

- **Smart Homes:** Monitors elders who live in home, automatically dials 911 if elder ill, falls

- **Smart buildings:** Senses presence of people, ambient temperature, people flow, dynamically adjusts heating/cooling

- **Smart Cities:** Real time data from Sensors embedded in street used to direct drivers to empty parking spots
References

- Android App Development for Beginners videos by Bucky Roberts (thenewboston)
- Ask A Dev, Android Wear: What Developers Need to Know, https://www.youtube.com/watch?v=zTS2NZpLyQg
- Busy Coder’s guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014