CS 403X Mobile and Ubiquitous Computing
Lecture 1: Introduction

Emmanuel Agu
About Me
A Little about me

- WPI Computer Science Professor
- Research interests:
  - mobile computing especially mobile health, computer graphics
- Started working in mobile computing in grad school
  - 3 years in wireless LAN research lab (*pre 802.11*)
- CS + ECE background (Hardware + software)
- Current active research: Mobile health apps
  - E.g: AlcoGait app to detect how drunk Smartphone owner is
Administrivia
Administrivia: Schedule

- **Week 1-3:** I will present (course introduction, Android programming, assigned projects)
  - **Goal:** Students acquire basic Android skills to do excellent project

- **Weeks 4 – 7:** Students will present papers
  - **Goal:** examine cutting edge research ideas
  - Student talks short and sweet (~15 minutes)
  - Discussions
  - Students not presenting submit summaries of any 1 of day’s papers

- **Week 4-7:** Final project
  - **Week 5:** Students propose final project
  - **Week 7:** Students present + submit final projects
Requirements to get a Grade

- **Seminar class:** Participate in class discussions (6%)
- **Weeks 4-7:** Student paper presentations (15%)
  - Each student will present 1 paper (in groups?)
- Students not presenting, submit summaries of any 1 of week’s papers (15%)
- **Projects:** 3 assigned (24%) and 1 final project(s) (40%)
- **Final project:** 5-phases (See website for deadlines)
  - Pick partner + decide project area
  - Brainstorm on ideas
  - Submit proposal intro + related work + proposed project plan
  - Build, evaluate, experiment, analyze results
  - Present results + submit final paper (in week 7)
- **Grading policy:** Presentation(s) 15%, Class participation 6%, Assigned Projects 24%, Final project: 40%, Summaries: 15%
Course Texts

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

- Will also use official Google Android documentation
- Research papers: Why not text?
Poll Question

• How many students:
  1. **Own** recent Android phones (running Android 4.4, 5.0 or 6.0?)
  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

- **Communication**: Talk, text, Internet access, chat
- **Computing**: Java apps, JVM, apps
  - Powerful processors: Quad core CPUs, GPUs
- **Sensors**: Camera, video, accelerometer, etc
- **Smartphone = Communication + Computing + Sensors**
- Google Nexus 5 phone: Quad core 2.5 GHz CPU, Adreno 330 GPU

<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 Quad-core</td>
<td>1.7 GHz Dual-core</td>
<td>1.3 GHz Dual-core</td>
<td>600 MHz</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>OpenGL ES 1.1</td>
<td>OpenGL ES 1.2</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
<td>OpenGL ES 2.0 Shader Model 4.1</td>
</tr>
<tr>
<td><strong>GFLOPS</strong></td>
<td>40-45</td>
<td>400 MHz</td>
<td>266 MHz (Tri-core)</td>
<td>200 MHz (1.6 GFLOPS)</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

Future sensors?
- Heart rate monitor,
- Activity sensor,
- Pollution sensor,
- etc
SmartPhone OS

- Over 80% of all phones sold are smartphones
- Android share 78% worldwide
- iOS 18%

Source: IDC, Strategy Analytics
Mobile Computing
mobile

*adjective*
/ˈmōbəl, ˈmō.bēl/

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

- Mobile? Human computes while moving
  - Continuous network connectivity,
  - Points of connection (e.g. cell towers) change
- **Note:** Human initiates all activity, (e.g launches apps)
- Network is mostly *passive*
- **Example:** Using *foursquare.com* on smart phone
What does mobile mean?

- Mobile computing = computing while location changes
- Location (e.g.) must be one of app/program’s inputs
- Different user location = different output (e.g. maps)
- User in California gets different map from user in Boston
What does mobile mean?

- Truly mobile 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 apps are not truly mobile?

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

[Images of Mobile banking app, Internet Retailer app, and Diet recording app]
Which of these apps are truly mobile?

a. Yahoo mail mobile

b. Uber app
Which of these apps are truly mobile?

c. Badoo dating app
Ubiquitous Computing
ubiquitous
\[\text{\textipa{yoo\-' bik\-w\-d\-ez}}]  

*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)
- 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
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 advise*: 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

**5 W’s + 1 H**
UbiComp Example: Moves App

- Counts Smartphone users steps through the day
Ubiquitous Computing: Wearable sensors for Health

remote patient monitoring

UbiComp: Wearables, BlueTooth Devices

- Body Worn
  Activity Trackers

- Bluetooth
  Wellness
  Devices

External sources of data for smartphone
A lot (Explosion) of Devices

- *Recent Nokia quote*: More cell phones than tooth brushes
- Many more sensors envisaged
- *Ubiquitous computing*: Many computers per person
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

- **Issues:**
  - Sensor design (miniaturization, low cost)
  - Smart spaces
  - Invisibility (room million sensors, minimal user distraction)
  - Localized scalability (more distant, less communication)
  - Uneven conditioning
  - Context-awareness (assist user based on current situation)
  - Cyber-foraging (servers augment mobile device)
  - Self-configuring networks
Sensor Processing

- **Machine learning** commonly used to process sensor data into higher level actions
  - Example: accelerometer data classified into user actions (walking, running, jumping, in car, etc)
Mobile CrowdSensing
Mobile CrowdSensing

- **Personal sensing:** phenomena pertain to individual
  - E.g: activity detection and logging for health monitoring

- **Group:** friends, co-workers, neighborhood
  - GarbageWatch to improve recycling, neighborhood surveillance

- **Community sensing (mobile crowdsensing):**
  - Many people contribute their individual readings
  - **Examples:** Traffic, air pollution, city noise maps, bike routes, fuel price
Mobile Crowd Sensing

- **Classic example:** Comparative shopping
- Compare price of toothpaste at CVS before buying

- **Example 2:** Waze crowdsourced traffic
Sense What?

- **Environmental**: pollution, water levels in a creek
- **Transportation**: traffic/road conditions, available parking
- **City infrastructure**: malfunctioning hydrants and traffic signs
- **Social**: photoblogging, share bike route quality, petrol price watch
- **Health and well-being**:
  - Share exercise data (amount, frequency, schedule),
  - share eating habits and pictures of food
Wireless Networks
Wireless Network Types

- Wi-Fi (802.11) (e.g. Starbucks Wi-Fi)
- Cellular networks (Wide area)
- Bluetooth
- Near Field Communications (NFC)
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