App Development for Smart Devices

CS 495/595 - Fall 2011

Tamer Nadeem
Dept. of Computer Science
Course Logistics
Welcome to CS 495/595

• Timings: Tu/Th 4:20pm to 5:35pm

• Location: KAUF 213 – only for Today
Dragas 1115 – starting next Tuesday

• Instructor: Tamer Nadeem
Ph.D from Univ. of Maryland, 2006
Research in Networks, Dist Sys, Mobile Comp.
Email: nadeem@cs.odu.edu
Office: E&CS Building 3204

• Office Hours: Tu/Th 2:30pm to 4:00pm, or by appointment
Welcome to CS 495/595

- **Prerequisites:** Comfortable with Java

- **Grading:**
  - Participation/Midterm: 30%
  - Programming Assignments: 40%
  - Final Project: 30%
Welcome to CS 495/595

• Class Webpage:
  • [http://www.cs.odu.edu/~cs495/](http://www.cs.odu.edu/~cs495/)
  • Please check course website frequently

• Make up classes:
  • Will be occasionally necessary due to travel
  • Fixed schedule versus case by case basis?
Welcome to CS 495/595

• Text:
  • Required: Ed Burnette, "Hello, Android: Introducing Google's Mobile Development Platform"
  • Recommended:
    • Reto Meier, "Professional Android 2 Application Development"
    • James Steele, Nelson To, "The Android Developer's Cookbook"
Welcome to CS 495/595

• Academic Integrity / Honor Code:

  • "I pledge to support the honor system of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member if the academic community, it is my responsibility to turn in all suspected violators of the honor system. I will report to Honor Council hearings if summoned."

  • Please refer to ODU Honor Council’s webpage: http://orgs.odu.edu/hc/
Welcome to CS 495/595

• Course Policy:
  • Grading:
    • 90-100  A
    • 80-89   B
    • 70-79   C
    • 0-69    F
  • Late assignments are not accepted.
  • Attendance
  • Email

Please refer to class webpage for more details.
Course Overview
This Course

• Introduces fundamentals of application development for Android phones

• Goals of this course:
  • Help you learn about mobile app development and best practices
  • Provide you with the tools, knowledge, and excuse to create a novel mobile app that helps solve a serious problem that strengthens your programming portfolio

• Envisions new practical mobile applications/services
Class Responsibilities

• I will lead lectures
  • You present 1 paper in entire semester (20 minutes)
  • 1-2 students presentation per class
  • Some classes will include coding

• For every class, read the readings list before the class

• Assignments should be on time
Course Structure

• 1 mid term?, No Final Exam
  • Tentative date of mid-term: Second half of Nov

• Semester-long class project
  • In groups of 2 (max 3).
    • Individual projects are allowed by permission
  • Focus on this from early on

• Class ends with a final project presentation & demo
Participation / Presentation

• Ask lots of questions. Period.
  • I strongly encourage you to ask, disagree, debate

• Class presentation
  • You present 1 paper (20 minutes)
  • Check class schedule by next week for reading papers
  • Email me any paper you are interested in
  • Pick an open slot (check class schedule)
    • Earlier you pick, more options you have to choose from
    • Deadline is Sep 06, 2011

• Email me your choice of paper (and date)
  • Don’t worry about not knowing the topic of paper
    • By that time, you will know enough
Thoughts on Reading Assignments

• Know why you are reading the paper
  • Reading for absorbing concepts (class assignment)
    • Read fully, think, reread, ask, challenge
  • Reading for excitement (deciding project topic)
    • Read initial parts, don’t try to understand everything, get a feel
Course Term Project

• Initial proposal due Oct 13 after Fall break
  • 1-page progress report every 2 weeks (due Friday night)
  • Final report + demo + presentation

• Projects consist of:
  • Application/Service identification
  • Solution design
  • Implementation

• Discuss your thoughts and ideas with me
  • They need not be cooked, and can have many flaws
  • Statistically, every 18 ideas lead to one decent idea

• If you like an area/direction
  • Search and Read many many related references
More on Projects

- Project ideas take time … think now and then
  - Spending 3 hours for 10 days better than 10 hours for 3 days

- Find a project partner(s) early
  - Search and discuss App/Services ideas

- Everyone in the class will try/critique apps from other teams

- At end of the course we will vote for the Top App

- Possible Application/Service domains:
  - Transportation
  - Education
  - Health
  - Energy
  - Smart Home
Labs/Facilities

• Development Environment:
  • Your laptop
  • E&CS 3104 (Open Research Lab)
  • SmartApp Lab (under construction)

• Collaboration:
  • BlackBoard Discussion (http://clt.odu.edu/bb/)
  • Internet
  • Friends/Colleagues
http://clt.odu.edu/bb/

Old Dominion University

My ODU Courses

201110 FALL CS495 25906: TPC: APP DEV FOR SMART DEVICES

My Blackboard & Courses

Communications

Annotations
- Teaching Staff
- Orientation
- Syllabus
- Schedule
- Modules
- Assignments
- Exams
- Help

Tool Box
- Communication
- Course Tools
- Course Map

Control Panel
- Refresh
- Detail View

Announcements

Collaboration

Messages

Roster

Discussion Board

Course Discussions: Dummy Forum

Thread Remove Collect Mark Read

9/1/11 10:24 AM Dummy Thread #1

CS 495/595 - App Development for Smart Devices
Questions?
INTRODUCTION
Mobile Computing

• Driven by technology and vision
  • wireless communication technology
  • global infrastructure
  • device miniaturization
  • mobile computing platforms

• The field is moving fast

• “People and their machines should be able to access information and communicate with easily and securely, in any medium each other or combination of media – voice, data, image, video, or multimedia – any time, anywhere, in a timely, cost-effective way.”, Dr. G. H. Heilmeier, Oct 1992

• “The mobile device will be the primary connection tool to the Internet for most people in the world in 2020.”, PEW Internet and American Life Project, Dec. 2008
Smart Devices

• A **smart device** is a device that is digital, active, computer networked, is user reconfigurable and that can operate to some extent autonomously.

• A **smart device** is a ubiquitous computing device: a device that exhibits some properties of ubiquitous computing including artificial intelligence.

• **Mark Weiser** categorized ubiquitous devices:
  • **Tabs**: accompanied or wearable centimeter sized devices, e.g., smartphones, smart cards
  • **Pads**: hand-held decimeter-sized devices, e.g., laptops
  • **Boards**: meter sized interactive display devices, e.g., horizontal surface computers and vertical smart boards.
Mobile Devices

- Multi-purpose ICT devices (e.g., personal office, mobile phone, camera, etc.)
- Mobility: loosely-bound vs. tightly-bound to users
- Personalized
- Operates as a single portal, e.g., a Web portal.
  - Internal application services
  - External services typically accessed local area wireless network
- Intermittent resource access
- A locus of control that resides in the smart device.
- Networked, distributed and transparently accessible.
- Context awareness
Smart Devices at Home/Office

- WiFi
- Satellite
- UWB
- Bluetooth
- Cellular

Connections:
- WiFi
- WiFi 802.11g/n
- Cellular
Smart Devices on Move

Source: http://www.ece.uah.edu/~jovanov/whrsms/
Smart Devices on Road

GSM/UMTS, cdmaOne/cdma2000, WLAN, GPS, DAB, TETRA, ...

road condition, weather, location-based services, emergency
Smart Devices for Intelligent Transportation

Mobile Millennium Traffic in San Francisco and the Bay Area

Source: http://traffic.berkeley.edu/

CarTel Project at MIT

Source: http://cartel.csail.mit.edu/doku.php
Smart Biomedical Systems

**In-body smart devices**
- sensors/monitoring devices
- drug delivery systems
- medical robots
- neural implants

**Wireless telemedicine**

**Recovery from nerve damage**
Habitat monitoring
Supply chain management
Mobile Social Networking
Smart Phone

Trend: everything in one small device
Smart Phone - Overview

- **A smartphone** is a high-end mobile phone

- **Combines the functions of**
  - mobile phone
  - personal digital assistant (PDA)
  - portable media players
  - camera phones
  - high-resolution touchscreens
  - web browsers
  - GPS navigation
  - Wi-Fi and mobile broadband access
  - etc.

- **Feature phone vs. Smartphone**

  - Feature phone
    - proprietary firmware & limited platforms

  - Smartphone
    - open and complete mobile operating system
    - tightly integrate with the user interface and phone features
    - relies on a more powerful application programming interface (API)
Smart Phone - History

Early Years:

IBM Simon (1992)

Nokia 9000 (1996)
Nokia 9110i (1998)
Nokia 9110i (2000)

Ericsson GS88 (1997)

Symbian:

Ericsson R380 (2000)
Ericsson P800 (2002)

Nokia 9210 (2000)
Nokia 9500 (2005)
Nokia E90 (2007)

Nokia N95 (2007)
Nokia N8 (2010)
Smart Phone - History

Palm, Windows, BlackBerry:

**Palm Kyocera 6035 (2001)**

**Windows CE Pocket PC (2002)**
**Windows Phones 7 (2007)**

**Palm OS Treo (2002)**

**RIM BlackBerry (2002)**

**iPhone:**

- iPhone (2007)
- iPhone 3G (2008)
- iPhone 4 (2010)

**Android:**

- Nexus One (2010)
Smart Phones - Statistics

Quantity Market Shares by Gartner (in one year) (New Sales)

<table>
<thead>
<tr>
<th>BRAND</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbian 2009</td>
<td>46.9%</td>
</tr>
<tr>
<td>Symbian 2010</td>
<td>37.6%</td>
</tr>
<tr>
<td>Android 2009</td>
<td>3.9%</td>
</tr>
<tr>
<td>Android 2010</td>
<td>22.7%</td>
</tr>
<tr>
<td>RIM 2009</td>
<td>19.9%</td>
</tr>
<tr>
<td>RIM 2010</td>
<td>16.0%</td>
</tr>
<tr>
<td>iPhone 2009</td>
<td>14.4%</td>
</tr>
<tr>
<td>iPhone 2010</td>
<td>15.7%</td>
</tr>
<tr>
<td>Microsoft 2009</td>
<td>8.7%</td>
</tr>
<tr>
<td>Microsoft 2010</td>
<td>4.2%</td>
</tr>
<tr>
<td>Other OS 2009</td>
<td>6.1%</td>
</tr>
<tr>
<td>Other OS 2010</td>
<td>3.8%</td>
</tr>
</tbody>
</table>

Notes: The table is for a whole year. In one year RIM is still above iPhone.

Share of worldwide 2011 Q2 smartphone sales to end users by operating system, according to Gartner.⁶⁵
Mobile Applications

• What are they?
  • Any application software that is developed for small low-power handheld devices such as personal digital assistants, enterprise digital assistants or mobile phones.

• Users on mobile phone’s
  • Typically check the news, weather, email, or their social networks
  • Often have a choice between the mobile web version or a specially-created mobile app.

• Mobile App Types
  • Web apps: run in a web browser
    • HTML, JavaScript, Flash, server-side components, etc.
  • Native: compiled binaries for the device
    • Not cross-platform, but more interesting options
Web App vs. Native App

The average Android user spends almost an hour per day interacting with web and apps

Proportion of Time Spent on Web vs. Apps
Nielsen Smartphone Analytics, June 2011

Source: Nielsen

56 Minutes per Day

Web 33%

Apps 67%
Native Development Environments

- Options
  - Java ME
  - .NET Compact Framework (C++, C#, VB.NET) for Windows Mobile
  - Qualcomm’s BREW (C or C++)
  - Symbian (C++)
  - BlackBerry (Java)
  - Android (Java)
  - iPhone (Objective-C)

Is having so many choices and so much industry turmoil/competition a good thing?
Development Environments

• Most platforms have an SDK that you can download and build against
• Every platform has an emulator that you can use to test your apps
• Most emulators are configurable to match a variety of mobile devices
  • Various screen sizes, memory limitations, tablets, etc.
  • In practice, emulators quite limited

IDE - integrated development environment that provides tools to allow a developer to write, test and deploy applications into the target platform environment.
xC ode IDE & iPhone Emulator
Eclipse and Android Emulator

Code snippet:
```java
package com.example.helloandroid;

import android.app.Activity;

public class HelloAndroid extends Activity {
    private static final String LOG_TAG = "HelloAndroid";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
```

Android Emulator interface:
- Google search bar
- Messaging app
- Phone app
- Contacts app
- Browser app
- Physical keyboard
- Soft keyboard
Smart Phone – the good

• Always with the user
• Increasingly powerful devices
• Typically GPS capable
• Typically have accelerometer
• Designed for communication
• 2+ types of wireless connections
• Many apps are free or low-cost
Smart Phone – the not-so-good

- Limited processor speed
- Limited RAM
- Limited battery life
- Limited, unreliable, and slow network access
- Limited screen size
- Limited permanent storage capacity
- Limited or awkward input
  - (none great: soft keyboard, phone keypad, touch screen, stylus, speech)
- Inconsistent platforms across devices
- High costs associated with data transfer
Android

- Android, Inc. founded in Palo Alto, California in October 2003
- Google acquired Android Inc. in August 2005
- Developed a mobile device platform powered by the Linux kernel
- Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system
- On November 2007, the Open Handset Alliance, a consortium of several companies (e.g., Broadcom, Google, HTC, Intel, etc. unveiled itself). The goal is to develop open standards for mobile devices.
- Open Handset Alliance unveiled their first product, Android, a mobile device platform built on the Linux kernel version 2.6
- Android OS (open source) released in October 2008
Why Android

• Simple and powerful SDK

• No licensing fees

• Excellent documentation, and a thriving developer community

• From commercial perspective
  
  • Requires no certification for becoming an Android developer
  
  • Provides the Android Market for distribution and monetization of your application
  
  • Has no approval process for application distribution
  
  • Gives you total control over your brand and access to the user’s home screen
Android Overview

• Open source OS and development platform
  • In theory, you can change anything
  • In practice....

• Hardware reference design

• Linux OS kernel

• Open-source libraries for app development
  • E.g., SQLite, Webkit, OpenGL, media manager

• SDK and tools

• Preinstalled apps

• Wild west of app stores: the Market
Android Version

- **Initial**: 1.5 (Cupcake), 1.6 (Donut)
- **2.0 (Eclair)**: new web browser, new user interface, support for HTML5, W3C Geolocation API, enhanced camera features
- **2.1 (Eclair)**: voice controls, 5 homescreens instead of 3, animated backgrounds,
- **2.2 (Froyo)**: speed improvements, Chrome V8 JavaScript engine, Wi-Fi hotspot tethering, Adobe Flash support
- **2.3 (Gingerbread)**: Near Field Communication
- **3.0 (Honeycomb)**: tablet-oriented release, supports multicore processors, hardware acceleration for graphics
- **3.1 (Honeycomb)**: directly transfer content from USB devices
- **3.2 (Honeycomb)**: adds several new capabilities for users and developer (e.g., providing developers with more precise control over the UI)
- **4.0 (Ice Cream Sandwich)**: combination of Gingerbread and Honeycomb
Features and Specifications I

- Platform is adaptable to larger, VGA, 2D graphics library, 3D OpenGL graphics library
- **Storage** - SQLite, a lightweight relational database
- **Connectivity** - supports connectivity technologies including GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Messaging** – SMS, MMS, threaded text messaging, Push Messaging service.
- **Multiple language support**
- **Web browser** - based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine.
- **Java support** – no Java Virtual Machine, Dalvik executables and run on Dalvik
Features and Specifications II

• **Media support** - audio/video/still media formats: WebM, H.263, H.264, MPEG-4 SP, WAV, JPEG, PNG, GIF, BMP, etc.

• **Streaming media support** - RTP/RTSP streaming (3GPP PSS, ISMA), HTML5 `<video>` tag, Adobe Flash Streaming (RTMP), HTTP Dynamic Streaming, Apple HTTP Live Streaming

• **Additional hardware support** - video/still cameras, touchscreens, GPS, accelerometers, gyroscopes, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits and accelerated 3D graphics

• **Multi-touch**

• **Bluetooth** - supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID)
Features and Specifications III

• **Video calling** – no native video calling, but some handsets have a customized version of the operating system that supports it. Video calling through Google Talk is available in Android 2.3.4 and later. Skype 2.1 offers video calling in Android 2.3, including front camera support.

• **Multitasking**

• **Voice based features** - Google search through voice and voice actions for calling, texting, navigation, etc.

• **Tethering**

**Google describes Android as:** “The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.”
Android SDK Features

- No licensing, distribution, or development fees
- Wi-Fi hardware access
- GSM, EDGE, and 3G networks for telephony or data transfer, allowing you to make or receive calls or SMS messages, or to send and retrieve data across networks
- Comprehensive APIs for location-based services such as GPS
- Full multimedia hardware control including playback and recording using the camera and microphone
- APIs for accelerometer and compass hardware
- IPC message passing
- Shared data stores
- An integrated open source WebKit-based browser
- Full support for app that integrate Map controls as part of their user interface
- Peer-to-peer (P2P) support using Google Talk
- Mobile-optimized hardware-accelerated graphics including a path-based 2D graphics library and support for 3D graphics using OpenGL ES
- Media libraries for playing and recording a variety of audio/video or image formats
- An application framework that encourages reuse of application components and the replacement of native applications
Android Development I

• “Open philosophy”
• Application framework that encourages reuse of app components
• Access to (much) hardware (sometimes even without bugs!)
• Release and fix mentality (or it seems like it, sometimes)
• Background services
• Shared data and inter-process communication
  • Notifications (via UI)
  • Intents
  • Content providers (managed access to app private data)
• All apps created equal
• Native Development Kit (NDK) (C++ tinkering under the hood)
• OS optimization of memory and process management
Android Development II

• Expect...
  • Limited processing power
  • Limited RAM
  • Limited permanent storage capacity
  • Small screen and low resolution
  • High cost of data transfer
  • Slow data transfer rates with high latency
  • Unreliable data connections

• Application framework that encourages reuse of application components
Android Development III

- “Screens” are Activities that are chained with lightweight exchange of data between them
- OS can handle stack of Activities if you want it to (e.g., back)
- OS Manages process lifetime (app assassin)

You MUST
- Ensure that your app is ready for swift death
- Yet, it must remain response and/or restart in the background
- Must come to the foreground quickly
Application Types

• Foreground
  • Useful when being used.
  • Suspended otherwise

• Background
  • Apart from when being configured, spends most of lifetime hidden (e.g., call screening app)

• Intermittent
  • Some interaction but mostly in the background (e.g., media player)

• Widget
  • Home screen status update
Application Good behavior

• Is well behaved
• Switches seamlessly from background to foreground
• Is polite (e.g., stealing focus)
• Presents a consistent user interface
• Is responsive
• Two conditions monitored
  • Must respond to any user action (e.g., key press) within 5s
  • A BroadcastReceiver must return from its OnReceive handler within 10s
Android Market (https://market.android.com/apps)

- Online software store developed by Google for Android devices
- As of December 2010 there were about 200,000 games, applications and widgets available on the Android Market.
- At end of June 2011 Google said there had been more than 6 billion Android apps installed
- The operating system itself is installed on 130 million total devices.
- Only devices that comply with Google's compatibility requirements are allowed to preinstall Google's closed-source Android Market app and access the Market.
- The Market filters the list of applications presented by the Market app to those that are compatible with the user's device,
Android Market

Most common SDK versions, August 25, 2011

- 1.5 (Cupcake)
- 1.6 (Donut)
- 2.1 (Eclair)
- 2.2 (Froyo)
- 2.3 (Gingerbread)
- 3.0/3.1 (Honeycomb)

AndroLib

New Applications & Games by Month (10^3)

- July 2009: 1,669
- January 2010: 3,888
- July 2010: 13,356
- January 2011: 33,356
- July 2011: 33,564

- July 2009:
- January 2010:
- July 2010:
- January 2011:
- July 2011:
Questions?
Assignment #1: Warming UP

• Due Fri Sep 2nd, 11:59pm
• Select three high-quality apps from the iPhone or Android Market that are related to one of the following application domains:
  • Increase driving safety
  • Traffic monitoring
  • Enhance education experience
  • Monitor/support personal health
  • Monitor/save energy consumption
  • Support smart environments
• Pick applications that are innovative in some way (e.g., use new phone functionality, easy-to-use, popular, or take an approach that is very different from competitor apps.
• Critically evaluate the three apps
Assignment #1: Warming UP

• For each app, create a one-page pdf write-up that includes the following:
  • Your name and the name of the app evaluated.
  • A summary paragraph that identifies key features of the app.
  • A paragraph or bullet list describing what makes this app better than competitors. Why did you select it?
  • A list of positive characteristics (e.g. high-quality graphics, fun, indispensable tool). Try to be as precise as possible about what makes it good. For example, don’t just say it is addictive; try to explain why it is addictive. Don’t just say that it “looks professional.” Explain what makes it look that way.
  • A list of negative characteristics (e.g. force close, slow, confusing menu titles). What could be done better? Be detailed
  • A paragraph identifying the target audience for the app. Who might use this?
  • A paragraph discussing what additional features and functionalities you would add to enhance it.

• Few samples will be selected “randomly” to be presented and discussed in the next class (be prepared).
Assignment #1: Warming UP

• Send me an ordered list of only 3 reports/articles/papers you mostly liked about phone applications/services/features!
  • Again use instincts/guts
  • Your presentation paper could be among this list
  • Where:
    • Internet
    • Conferences: ACM Mobicom, MobiSys, HotMobile, IEEE SECON
    • Magazines

• Submission Format:
  • Send all pages in a single PDF file
  • Have a cover page with your name and your email
  • Send the file as an attachment to me
  • Make the subject line of the email: cs495_assignment_1