COMP5216 Mobile Computing (2015-Semester-2) REVIEW

# MOBILE COMPUTING - INTRODUCTION

[IT history towards mobile computing](https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/mobile-computing)

*The history of mobile computing can be divided into a number of eras*

1. *hardware: portability, miniaturization (size), converge functions such as touch screen, camera, etc. diverge functions to specific devices, songs - ipod, game - psp.*
2. *software: new operating system, Apps - App Store*
3. *internet: provide connection between equipments*
4. *web: 1.0 (concentrate on contents) + 2.0 (user-generated contents, interaction)*
5. *mobile (wearable): phone, bands*

[Definition of mobile computing](https://en.wikipedia.org/wiki/Mobile_computing)

*Mobile Computing is a technology that allows* ***transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link.*** *The main concept involves:*

1. *Mobile communication*
2. *Mobile hardware*
3. *Mobile software*

[Ecosystem of mobile computing](https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/mobile-computing)

*Mobile computing plays a central role in concert with other ubiquitous computing resources. Most people use multiple mobile devices for different purposes, but they also use a multitude of stationary or embedded computer systems, at work, at home, in their cars, or in the city around them. In concert, this makes up a rich digital ecosystem of interactive devices, systems and services often referred to as ubiquitous or pervasive computing, in which mobile computing is a central, but not the only, component.*

Impact, potential, challenges of mobile computing

1. *Impact:*

*Dressing/cloth, eating/food, living/accommodation, transport*

1. *Potential:*

*Commerce and Business, Communication, Education (anytime, anywhere, anybody), Entertainment, Tourism, Health*

1. *Challenges:*

*Search (context, efficiency), Personal data (manage, secure, share), Resources (power, storage...)*

# MOBILE PROGRAMMING

[Background and history](http://www.slideshare.net/emmaroberts477/history-of-the-development-of-mobile-applications)

*Android:*

* + - *Founded in 2003 by Andy Rubin Based on Linux kernel*
    - *Acquired by Google in 2005*
    - *T-Mobile G1 (Android 1.0), 2008*

*IOS*

* + - *iOS 1.x, 2007*

*Windows*

* + - *Windows Phone 7, 2010*
    - *Windows 1.0, 1985*

Development cycle - practical steps for building apps

* + Scoping; UI design; coding; publish & maintain
    - *Scoping*
      * *The questions we asked before.*
      * *The main functionalities of the app*
      * *Useful*
      * *Unique or better The ~wow~ factor*
    - *UI design*
      * *Design prototyping*
      * *InVision - free forever*
      * *http://www.invisionapp.com/ Pinterest - Get inspired*
    - *Coding*
    - *Publish and maintain*
      * *Digitally signed*
      * *Approval process*
      * *Add new features*
      * *Fix bugs and clumsy designs*

[Android app architecture](http://www.tutorialspoint.com/android/android_architecture.htm)

[Development environment](https://developer.android.com/sdk/index.html)

[Brief API Guide](#_4d34og8)

* + [Activity](http://developer.android.com/reference/android/app/Activity.html); [service](http://developer.android.com/reference/android/app/Service.html); [content providers](http://developer.android.com/guide/topics/providers/content-providers.html); [broadcast receiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html); [intent](http://developer.android.com/reference/android/content/Intent.html);
  + [UI components](http://developer.android.com/guide/topics/ui/overview.html); [UI layouts](http://developer.android.com/guide/topics/ui/declaring-layout.html); [material design](http://developer.android.com/design/material/index.html); [data persistence](http://developer.android.com/guide/topics/data/data-storage.html);
  + [Cloud service](https://cloud.google.com/solutions/mobile/how-to-build-mobile-app-with-app-engine-backend-tutorial/); [multimedia](http://developer.android.com/guide/topics/media/mediaplayer.html); [sensors](http://developer.android.com/guide/topics/sensors/sensors_overview.html); other features

[Development strategies for cross-platform mobile app](http://www.anubavam.com/cross-platform-mobile-development)

M-V-VM [[1]](http://www.codeproject.com/Articles/100175/Model-View-ViewModel-MVVM-Explained) [[2]](https://msdn.microsoft.com/en-us/library/hh848246.aspx) [[3]](https://msdn.microsoft.com/en-us/library/hh821028.aspx)

[IoT and Windows app development](https://dev.windows.com/en-US/iot)

Mobile innovation

* + Real innovation is about solving problems, not having idea

# CLOUD COMPUTING

[History of computing](http://www.computerweekly.com/feature/A-history-of-cloud-computing)

* Different computing paradigms [[1]](https://en.wikipedia.org/wiki/Cloud_computing)

[Cloud services](https://en.wikipedia.org/wiki/Cloud_computing#Service_models)

[Challenges of cloud computing](#_44sinio)

[Future directions of cloud computing](#_2jxsxqh)

[Mobile cloud computing](https://en.wikipedia.org/wiki/Mobile_cloud_computing)

# MOBILE COMMERCE

Definition of E/M/L/S-Commerce

1. *E-Commerce (Electronic Commerce)*

*Exchange of goods, services, and money among firms, between firms and their*

*customers, and between customers, supported by communication technologies,*

*in particular, the Internet*

1. *M-Commerce (Mobile Commerce)*

*E-Commerce transactions that are conducted in a wireless environment*

1. *L-Commerce (Location-based M-Commerce)*

*M-commerce transactions targeted to individuals in specific locations at specific*

*times*

1. *S-Commerce (Social Commerce)*

E-commerce activities and transactions through social media

Different types of mobile applications for commerce

1. *Mobile empowering*

*Mobile Payments 移动支付 - payment infrastructure, mobile POS, direct mobile payment*

*Retail Enablement 零售支援 - contextual commerce, in-store marketing, mobile couponing, location-based offers*

1. *Mobile enhanced*

*Mobile Retail 移动零售- traditional commerce, flash sales & daily deals, subscription commerce, vertical brands*

*Market places 交易平台- consumer goods, real estate & apartments, short term stays, office space, caregivers & doctors, car sales, car sharing, dining experience*

1. *Mobile enabled*

*On-demand services 按需服务- ridesharing, meals, groceries, laundry, home care & cleaning, self storage, valet parking*

*App-based services 基于APP服务- dating, education, personal finance, productivity, fitness, healthcare, nutrition, mental health*

Mobile payments

*Direct Operator (Carrier) Billing (e.g Bango)*

*Payment is billed to the mobile account*

*Digital Wallets (e.g PayPal)*

*Program or web service that stores all shopping information in one central place*

*Mobile Proximity Payment (e.g PayPass)*

*-Contactless payment for purchases made in physical stores*

*-Near Field Communication*

Trend: Social-Location-Mobile commerce

*The integration of social networking platforms, local context (location data)*

*and mobile technologies*

*Social - Awareness, Familiarity, Opinion/Imagery*

*Mobile - Consideration, One make/Model intention*

*Location - Shopping, Purchase*

-Applications:

Marketing research, Communication, Sales promotions & discount, Relationship development and loyalty programs

-Opportunities:

Add convenience, Provide unique value, Provide social value, Offer incentives, Entertain

# MOBILE HEALTH

Definition of teleHealth

Use of telecommunications for the direct provision of care to patients at a distance

Using telecommunications to communicate with patients and deliver services such as

– Audio-Visual consultations e.g., Nursing homes (outpatient follow-up), Interpretations

– Patient data monitoring (EMR) e.g., Home Dialysis for automated alerts

– Patient portals e.g., http://health.gov.au/

– Scheduling appointments e.g., online booking systems

– Remote image viewing (Teleradiology)

– Remote access of EHR

Telehealth challenges

– Complexity of infrastructure

– Degree of integration

– Message structure

– Cost

– Security

– Reimbursement

Mobile health - wellbeing and wearables

Mobile apps in healthcare

# MOBILE GAME

Building a game

• Game

• Ideas

• Tools

• Building a game

• What type of game

• Revenue

• Platforms

• Pipeline

• Performance

• Where to start

Ideas and tools

Ideas

• Start simple

• Start very simple

• Head to a Game Jam

• Join IGDA and head to meetups

Tools

• Unreal Engine

• Unity Engine

• Cocos2D

• Gamemaker

• ... and more

Pipeline and performance

Pipeline

Design->Implement->Test->Design

Performance

• File Size

• Image Size

• Audio

• Sensors

• Touch

• Accelerometer

• Compass

• Camera

• Microphone

• Analytics

• Draw calls

• Shader capabilities

Revenue models

PAID

PREMIUM

IN APP PURCHASE

FEATURE LIMITED

SUBSCRIPTION

TIME LIMITED

FREE

SPONSORED

AFFILIATE

AD SUPPORTED

LITE

FREE

# APPENDIX

## Activity

􄡦 Activity: An activity represents a single screen with a user interface.

􄡦 public class Activity extends ApplicationContext {

protected void onCreate(Bundle savedInstanceState);

protected void onStart();

protected void onRestart();

protected void onResume();

protected void onPause();

protected void onStop();

protected void onDestroy();

}

􄡦 onCreate(): Called when the activity is first created. This is where you should do all of your normal static setup: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one.

􄡦 onRestart(): Called after your activity has been stopped, prior to it being started again. Always followed by onStart().

􄡦 onStart(): Called when the activity is becoming visible to the user. Followed by onResume() if the activity comes to the foreground, or onStop() if it becomes hidden.

􄡦 onResume(): Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it. Always followed by onPause().

􄡦 onPause(): Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns.

􄡦 Followed by either onResume() if the activity returns back to the front, or onStop() if it becomes invisible to the user.

## Services

􄡦 Services: A service is a component that runs in the background to perform long-running operations or to perform work for remote processes.

## Content Providers

􄡦 Content providers: A content provider manages a shared set of app data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your app can access.

## Broadcast Receivers

􄡦 Broadcast receivers: A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured.

## Intent

􄡦 Intent: An intent is a messaging object you can use to request an action from another app component. You can use intents to: Start an activity; Start a service; Start a broadcast.

## UI Components

􄡦 To declare your layout, you can instantiate View objects in code and start building a tree, but the easiest and most effective way to define your layout is with an XML file. XML offers a human-readable structure for the layout, similar to HTML.

## UI Layouts

􄡦 LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute.

􄡦 RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).

## Other Features

􄡦 Bluetooth, iBeacon; 􄡦 Gyro sensor (vibration); 􄡦 Image and video processing

􄡦 Video streaming; 􄡦 Gesture detection; 􄡦 Canvas drawing; 􄡦 3D graphics; 􄡦 etc…

## Challenges of Cloud Computing

1: Efficiency of service provisioning

2: Sustainability

3: Effectiveness of service usage and control

4. Transparency of service delivery and billing

5. Information security

6. Data privacy

7. Interoperability

8. Portability between providers

9. Ensuring fair competition in the market

10. Compliance with regulatory requirements

## Future Directions of Cloud Computing

1. Cloud storage

2. Mobile clouds

3. Hybrid clouds

4. Service brokers

5. Cloud-centric design

6. Decision-making business frameworks

7. Cloud-based data centers