The Joys and Sorrows of Teaching and Developing Apps for Mobile Devices
or: how to become a workaholic, earn no money and live happily ever after

Davide Martinenghi

Rome, July 2, 2014
Outline

- Numbers and players of today’s mobile world
- Developing & producing an app
- Teaching “Mobile Applications”
- Marketing an app
Diffusion of smartphones: trend

- The number of smartphone subscribers has increased
  - +29 percent from a year ago (US)
  - +99 percent from two years ago (US)

Source: 2013 Mobile Future in Focus by Lipsman&Aquino
Diffusion of smartphones: volume

- devices sold by the billion per year

Source: IDC
Diffusion of smartphones: players

Source: Gartner
Market share

World-Wide Smartphone Sales (%)

Source: Gartner
Using a phone ≠ buying a phone

- Today: two main players (maybe three?):
  - iOS
  - Android
  - (Windows Phone is emerging)

Mobile operating system browsing statistics on Net Applications

- iOS: 48.34%
- Android: 41.58%
- Java ME: 3.46%
- Symbian: 2.52%
- Windows Phone: 2.10%
- BlackBerry: 1.14%
- Kindle: 0.72%
- Other: 0.14%

May 2014
# iOS vs Android

<table>
<thead>
<tr>
<th>Supported brands</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Apple)</td>
<td></td>
<td>dozens</td>
</tr>
<tr>
<td>bra</td>
<td>3Q, 7-11, 7251, Abocom, Acer..., Samsung, ..., ZTE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supported devices</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>A few handfuls</td>
<td></td>
<td>Hundreds</td>
</tr>
<tr>
<td>iPhone 1,3G,3GS,4,4s,5,5s, 5c; iPod Touch 1,2,3,4,5; iPad 1,2,3,4,Air; iPad mini 1,2; Apple TV 2,3,3RA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed source, proprietary</td>
<td></td>
<td>Open source, free (mostly)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price range</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>high level</td>
<td></td>
<td>all levels</td>
</tr>
</tbody>
</table>

- **Different types of users!**
User types: iOS vs Android

- iOS users are (slightly) younger than Android users

- iOS users
  - are more engaged in all categories
  - have higher income
    - (41% earn $100,000+ vs 24% among Android users)
  - are more likely to use M-Commerce (paid apps, in-app purchases)
  - show higher device loyalty

- Android users
  - are more numerous in all categories
Top grossing apps: a few examples

- **Clash of clans (Supercell):**
  - Daily: $1M+, 34K+ installs, 4M DAU
  - free with in-app purchase
  - $5.19 Average Revenue Per User (ARPU)

- **Candy Crush Saga (Zynga)**
  - Daily: $900K+, 26K+ installs, 7M DAU
  - free with in-app purchase

- **Minecraft – Pocket Edition**
  - #1 paid app, $50K per day

- **Ruzzle (MAG Interactive)**
  - Daily: 100M matches played (peak)
  - free + ads or paid
  - average eCPM: 4$, income estimate: $400K per day

Sources: thinkgaming.com – data for iOS alone
Business models

- Free
- Free + ads
- Paid (Premium)
- Free + in-app purchase (Freemium)
- Paid + in-app purchase

Apple App Store - United States
November 2013

Apple App Store - Japan
November 2013

Nearly the only model that counts
Ads: main indicators

- Classical pattern:
  - Lite app with ads -> Full app ad-free

- Ads are chosen in real time through bidding with several involved parties (ad networks, mediators, integrators)

- Effective Cost Per Mille (eCPM): main indicator for ads
  \[
  \text{eCPM} = \frac{\text{Total Earnings}}{\text{Total Impressions}} \times 1,000
  \]

- But also:
  - CPD: cost per download
  - CPI: cost per install
  - CPV: cost per view
  - CPC: cost per click

- All these vary and depend on specific events, campaigns, your click-through rate (CTR)…
## Ads: market share

**All Ad Networks - All Countries - Nov 28, 2013 ~ Feb 25, 2014**

### Ad Revenue Table

<table>
<thead>
<tr>
<th>Ad Network</th>
<th>Ad Revenue</th>
<th>eCPM</th>
<th>eCPC</th>
<th>Impressions</th>
<th>Clicks</th>
<th>Fill Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>$711,899.86</td>
<td>$30.31</td>
<td>$0.31</td>
<td>23,684,617</td>
<td>2,287,138</td>
<td></td>
</tr>
<tr>
<td>AdMob</td>
<td>$189,033.49</td>
<td>$30.24</td>
<td>$0.31</td>
<td>6,251,688</td>
<td>601,417</td>
<td>66%</td>
</tr>
<tr>
<td>Chartboost</td>
<td>$135,250.48</td>
<td>$30.54</td>
<td>$0.31</td>
<td>4,428,485</td>
<td>431,873</td>
<td>66%</td>
</tr>
<tr>
<td>iAd</td>
<td>$82,384.24</td>
<td>$30.71</td>
<td>$0.32</td>
<td>2,683,047</td>
<td>254,925</td>
<td>68%</td>
</tr>
<tr>
<td>Jumptap</td>
<td>$81,448.49</td>
<td>$30.28</td>
<td>$0.32</td>
<td>2,690,257</td>
<td>257,603</td>
<td>67%</td>
</tr>
<tr>
<td>Tapitl</td>
<td>$78,105.29</td>
<td>$30.75</td>
<td>$0.30</td>
<td>2,540,159</td>
<td>258,862</td>
<td>65%</td>
</tr>
<tr>
<td>Tapjoy</td>
<td>$77,882.92</td>
<td>$29.80</td>
<td>$0.31</td>
<td>2,613,316</td>
<td>254,410</td>
<td>66%</td>
</tr>
<tr>
<td>MdotM</td>
<td>$73,794.96</td>
<td>$29.78</td>
<td>$0.32</td>
<td>2,477,665</td>
<td>228,248</td>
<td>66%</td>
</tr>
</tbody>
</table>
The ads zoo

- Many different ad formats:
  - Banners
  - Expandable banners
  - Interactive banners
  - Interstitials
  - Videos
  - Floating banners

- A huge and quickly changing set of ad providers
  - iAd by Apple
  - AdMob by Google
  - MobFox
  - Vungle
  - Komli Media
  - Smaato
  - The MoPub
  - Mirosoft adCenter
  - AppFlood
  - ...
App market: a dangerous place to be

- **Candy Crush**
  - King owns the “candy” trademark (a common word!) in the EU for both games and clothing
  - Application for trademark abandoned in the US after acquiring Candy Crusher
  - Releasing an app with a name similar to “Candy Crush” will cause legal attack by heavy artillery
  - Similar attempts with the word “saga” to protect Candy Crush Saga

- **The Tetris Company**
  - All clones of Tetris get kicked out of the stores
    - Even though gameplay cannot be patented
  - Can’t use names that recall the name “Tetris”
    - Even though the word comes from a scientific term designating a geometric shape (tetromino)
How many apps in the stores?

- **Android:**
  - more than 1 million apps
  - over 25 billion downloads

- **iOS:**
  - more than 1.2 million apps (nearly half are iPad native)
  - over 75 billion downloads
  - 500+ new apps daily

Sources: mashable.com, TechCrunch
Developing apps

- Two main options:
  - Native development
  - Crossplatform development

- Go native if you want to
  - achieve native look and feel
  - achieve maximum performance
  - have the latest functionalities
  - get in the app stores

- Go cross-platform if you want to
  - quickly prototype a service or an actual app
  - have something that looks like an app based on HTML5
  - still have access to the phone hardware (as opposed to web)
Native applications

- **Advantages**
  - Native graphical interface
  - Better user experience
  - Best performance
  - Access to all HW & SW resources
  - Compatibility with all devices
  - App store available

- **Disadvantages**
  - Need to develop a version of the app for every platform
  - Knowledge of platform-specific programming language required
  - Longer times and higher development costs

Now also Swift…
Cross-platform applications

Types:
- Web
- Hybrid
- Interpreted
- Cross-Compiled

Advantages
- Unique development for several platforms
- Knowledge of platform-specific programming languages not required
- Use of Web programming languages (reusable GUI)
- Shorter times and lower development costs

Disadvantages
- Limited user experience
- Limited performance
- Limited access to HW & SW resources
- Compatibility problems with the different devices (and complex debugging)
- No target app store
Available cross-platform frameworks and tools

- **Web**
  - AppsBuilder
  - iBuildApp

- **Hybrid (thin wrapper around mobile Web browser)**
  - PhoneGap (HTML/Javascript/CSS)
  - MoSync (HTML/Javascript/CSS and C++)

- **Interpreted (abstraction layer, achieve native look&feel)**
  - Appcelerator Titanium (Javascript)
  - Rhodes (Ruby)

- **Cross-compiled**
  - Xamarin (C++)
  - Corona (Lua)
Gaming frameworks

- Dozens of different game engines
- Many underlying technologies (OpenGL, WebGL, DirectX, ...)

![Cocos2D](image1.png)

![Unreal Technology](image2.png)

![Unity](image3.png)
Many tools

- **iOS:**
  - XCode

- **Android:**
  - Android Studio
  - Eclipse
Many languages, quickly changing

- **iOS:**
  - Objective-C
    - modern syntax
    - memory management (manual RC -> ARC)
  - Swift
    - out in October
    - with the new OS and phones
  - Libraries (called frameworks) change at every new version of iOS (deprecation)
  - New version of iOS every year

- **Android:**
  - Java
  - Use of a specialized JVM (Dalvik)
  - No AWT or Swing
Teaching mobile apps: what to teach?

- too many platforms, tools, languages, devices, frameworks
  - can’t teach them all
  - can’t know them all

- go deep or go broad?

- one extreme: choose one platform and go deep
  - Example: Stanford, iOS, objective C, Xcode, Mac
  - Macs are expensive for students, the average uni can’t afford it

- other extreme: don’t choose and give general considerations about mobile apps
  - not so many general considerations after all
  - a few general patterns (e.g., MVC), but not enough to cover one course
  - things change too fast anyways – general observations from last year no longer apply (limitations, memory, speed, …)

- BTW: can’t recycle your material from a year to the next
Teaching mobile apps: how to teach?

- Students are (seemingly) interested in knowing these topics (some of them genuinely are!)
  - (alas, the grade is the only thing that really matters to them)

- They need to do something practical (an app!)

- They need to have a line in their cv for the job market

- Project course? Lab?

- Do all the students meet the minimum requirements?
  - Fluency in OOP should be mandatory

- Will they have the time to develop a project?

- Should they work in groups? How big?

- Should they be supervised during the project?

- Should the exam include a theory part?
The case Mobile Applications 2013: setting

- 80 students, Polimi, in Como, master’s program
- several with little programming experience
- course type: teaching + project
- covered iOS (13 hours), Android (10 hours), cross-platform tools (10 hours), plus general stuff and seminars from professional app developers

project:
- platform chosen by the students
- subject chosen by the students, approved by the teacher
- submitted in three phases (proposal, presentation, code)
- developed in groups of 1 to 3 people

mark:
- 50% determined by the app
- 50% determined by discussion during the exam
The case Mobile Applications 2013: outcome

- 4-5 apps were really excellent
  - some were unbelievably bad...
  - after 3 exam calls, less than 40% even tried the exam

- Many attempts of scam
  - several projects in which only one person in the group did the job
  - some projects in which *no one* in the group did the job
  - some project descriptions taken verbatim from the app store

- Difficult to evaluate a project
  - even when it’s easy to evaluate
  - all the students think their app is the best in the world
  - (or pretend so)
  - difficult (maybe?) to compare different apps
The case of Mobile Applications 2013: issues

- Major attendance dropout
  - choice of platform/subject was midways
  - after that, students stopped attending classes regarding other platforms
  - down to 2-3 students for less attractive platforms

- Students found the course difficult
  - They had to learn how to program
  - They had to learn how to program for mobile devices
  - They felt they had no time to do that
Mobile Applications: a proposal for 2014

- Still keep a project (maybe 2), but aim much much lower
  - The teacher choose the subject(s)
  - Keep it very simple (few hours programming)
  - Only one delivery date for all the students, before the first exam call

- Exam:
  - App + discussion is only worth 50% of the score
  - Those who did not deliver the app will have to do some (hard) old-day coding on paper during the exam
  - The remaining 50% is a traditional written exam with theory questions

- This should kill scammers
  - and kill excellences too
  - but excellences can express themselves during the thesis
A case study: a verticalized casual game

- The idea:
  - Take a simple, successful game, highly cloned, highly played (word game)
  - Adapt it specifically for a new market (Thailand)

- The process:
  - Choose a target platform (iOS)
  - Develop the concept (a few hours programming)
  - Acquire distribution license (a few days)
  - Make the app market-ready (several weeks)
  - Decide availability and model (ads, cost, device, OS)
  - Publish the app (at least one week to get approval)
  - Monitor app’s progress (downloads, revenue, DAU, ...)

- The cost:
  - $99 for the license
  - A few bucks here and there for pngs and sounds
  - Small percentages on revenue for few collaborators
  - Your free time
A case study: more on the process

- Develop a proper user interface (weeks)
  - From a one view multiple tabs and storyboards
  - Personalization (avatar, settings...)
  - Find partners for design, sound and beautification
  - Bilingual menu (Thai and English)

- Integrate ads (days/weeks to get online)

- Make the app social (in the hope it gets viral)

- Find out relevant legal aspects

- Configure the app on iTunesConnect
  - the first time, it takes the patience of a blessed

- Prepare distribution certificates on MemberCenter
  - it takes the patience of a saint, nondeterminism

- Use Game Center features
  - no additional cost (using servers in the cloud has a cost)
  - permanent leaderboards and achievements
  - online gaming: a bloodbath (nondeterminism, again)
A case study: the trends

Featured by Apple in some category

Installs

Cumulative revenue ($) by ads

DAU
A case study: the trends

Online gaming out

Installs

All Updates (iOS)

update
A case study: some figures

- 11K+ installs in 5 months
  - 50 to 150 per day
  - only nearly 50% of them use Game Center (5.7K so far)
  - only nearly 3K are still using the app (customer attrition)
  - 1M iPhones estimated in Thailand
  - we covered 1%: is it good? is it growing?

- 250+ Daily Active Users

- 1000+ ad views per day
  - eCPM: avg = 2.49, min = 0, max = 32
  - never believe ad networks claiming eCPM > 4

- 4.5 stars average rating on the App Store
  - but just over 50 ratings, little significance
  - less than 0.5% of customers rates an app
  - reviews are about 50% of the ratings

- $1.2 average daily income (min = $0, max = $7.8)
A case study: multipliers

- Multiply the supported devices
  - add iPad (supported but not visible: 1% of users)
  - requires adapting (and maintaining UI once more)

- Multiply the modes
  - ad-free paid app (effortless)
  - in-app content (boosters and cheats of some sort or extra art, requires some thinking and coding)

- Multiply the languages/countries/cultures
  - more than internationalization of menu
  - requires adapting gameplay to a new language
  - not always possible, moderate effort

- Multiply the platforms (Android, Windows Phone?)
  - big effort, not worthwhile at this level of income
  - but might ignite a viral effect
Discussion

- Mobile apps are
  - easy to use and understand
  - difficult to develop and to teach
  - dangerous to develop and to teach
  - students will hate you, you will hate them (but they are younger, more cunning and aggressive...)

- Indie app development is penniless
  - every minute spent promoting an app or finding an investor for an app is much better spent than a minute spent programming an app
  - but developing mobile apps can be fun
  - earning some money out of it can take a long time (or forever) but it’s not impossible
More discussion...?
Acknowledgments:
CUbRIK Project

- CUbRIK is a research project financed by the European Union

Goals:
- Advance the architecture of multimedia search
- Exploit the human contribution in multimedia search
- Use open-source components provided by the community
- Start up a search business ecosystem

http://www.cubrikproject.eu/