

Sviluppo Mobile

Introduzione

Lezioni

- Martedì 16-19
- Giovedì 16-19

Durata

- 4 settimane (corso Android)
- 4 settimane (corso iOS)

Materiali

- <https://github.com/2amcode/SviluppoMobile2014>

Storia

- Fine anni 90 piccoli giochi arcade, editor di suonerie
- Applicazioni sviluppate dai costruttori
- Applet Java di crescente complessità

WAP!

- Wireless Application Protocol
- Versione limitata di HTTP per far fronte ai limiti di memoria e di banda dei primi cellulari
- WML, pagine più semplici in design

Platform Wars

- Blackberry
- Palm OS
- Java ME
- Symbian OS (Nokia, Sony Ericsson, Motorola e Samsung)
- Apple
- Android

Apple

- 2007: iphone con “web apps”
- Piattaforma proprietaria
- Inizialmente chiusa (con eccezione di web apps)
- Jailbreak e prime applicazioni
- Cocoa Touch SDK Rilasciata nel 2008
- App Store

Android

- Android Inc.
- Rilasciato nel 2008
- Open Source
- Java come linguaggio
- Bytecode compilato

L'App Store e l'Android Market

- Sviluppatori "indie"
- Facilità di distribuzione a prezzi accessibili
- Mercato mondiale
- Costi contenuti (specialmente giochi)

Le “Apps”

- Comunicazione immediata
- Interfaccia accattivante
- Social
- Approccio “emozionale”

Analogie sviluppo desktop

- Interfaccia verso l'utente
- Dati e persistenza
- Produttività
- Giochi

Differenze sviluppo desktop

- Approccio multiutente
- Social
- Accesso file system
- Sandbox
- Accesso risorse hardware

Differenze sviluppo desktop

- Sensori
- Posizione
- Hardware a basso livello
- Uso ovunque, ma limitato nel tempo

Analogie nello sviluppo mobile

- Memoria limitata
- Banda limitata
- Ciclo di vita dell'applicazione
- Riduzione asset applicativi
- WebView
- Intent e URL Schema

Differenze nello sviluppo mobile

- Layout
- Emulazione
- Sandboxes e condivisione
- Multitasking
- Sicurezza
- Processo di review

UX: analogie

- Liste
- Dettagli
- Componenti UI: tabs, sliders, checkbox, switches
- Gestures: tap, swipe, pinch, drag

UX: differenze

- Schermi differenti, risoluzioni differenti
- Back in iOs, “up” e “back” in android
- Data views: segmented control vs spinner
- Action sheet vs radio buttons
- One size doesn't fit all

Sviluppo ibrido

- Adobe Phonegap
- IBM Worklight
- Titanium
- Apportable

Pro sviluppo ibrido

- Multipiattaforma
- Linguaggi web (HTML5, Javascript, CSS)

Contro sviluppo ibrido

- Performances
- Esperienza utente
- Accelerazione HW
- Webview non ottimizzate

Highlights

- Schermo piccolo
- Memoria limitata
- Esperienza utente / grafica
- Gestione sensori
- Fruizione vs. creazione

Android: evoluzione contesto

- 2003: Android inc
- 2005: acquisizione da parte di Google → visione del futuro mobile e necessità di applicare il business della pubblicità in ambito mobile (es: facebook e difficoltà a monetizzare su applicazioni mobili)
- 2007: fenomeno iPhone
- 2008: HTC Dream (Android 1.6)

Android: punti di Forza

- Natura "open source"
- - "gratuito", importante per i produttori di dispositivi
- - personalizzabile (HTC sense)
- Sviluppo in Java
- - ereditato una comunità e strumenti di sviluppo
- - utilizzo API Java
- Base Linux
- - kernel (dipende da versione Android e OEM)



Android: criticità

- Natura open
- - elevata frammentazione (gestione molti dispositivi e piattaforme)
- - difficoltà gestione aggiornamenti
- Java (bontà virtual machine...vedi ART)
- Strumenti di sviluppo meno evoluti in relazione ad iOS (Android studio)
- Efficienza generalmente minore
- Alcune discutibili scelte (o dati di fatto) riguardo librerie/funzionalità

Buon lavoro!