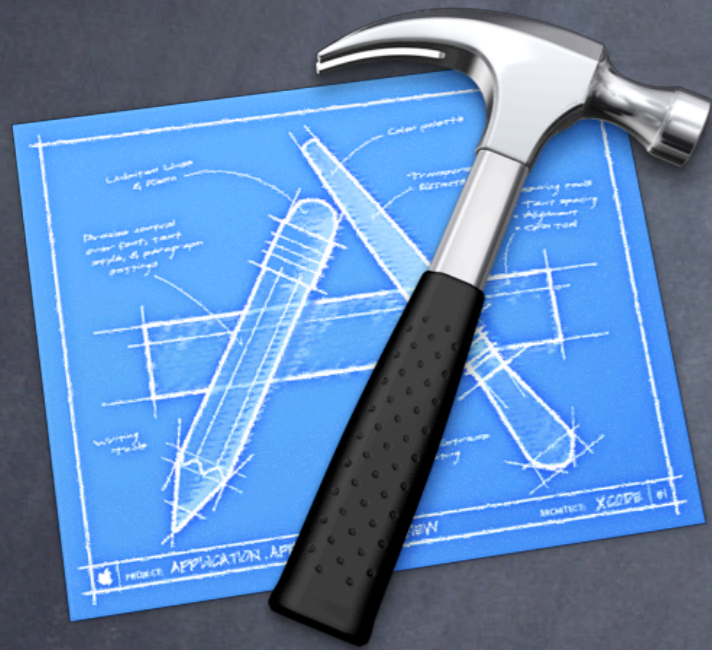


Why iPhone Changes Everything ... and how



+



=

users
impact
\$\$\$\$

Ben Bederson

Human-Computer Interaction Lab

University of Maryland

The Platform

iPhone	others
Large 320x480 screen	Smaller or lower-res screens
600 MHz CPU	Max 400 MHz
GPU w/ OpenGL	None (mostly)
Accelerometer	None
High quality touch screen	Pathetic wrongness
WiFi	mix
Real web	mix

The Platform

iPhone	others
AppStore	coming
They own music	wannabes
Consistent platforms	Wide variety
Soup to nuts	Ecosystem of software & hardware

Living a Mac Life

- Excellence and wonder, but ...
- Trade-offs, trade-offs, trade-offs
- Mac + VMWare Fusion = goodness
- Office lovers won't leave Office behind

Beyond today's scope

The Industry

Platform	Language	# Devices
Apple iPhone	Objective C	~10 million
Microsoft WinMo	C# (.NET)	~20 million
RIM BlackBerry	Java	~40 million
Nokia Series 60 (Symbian)	C++	???
Google Android	Java	0 (wait a week)

The Use

- iPhone sales eclipsing other platforms and accelerating
- 150 million old iPod owners upgrade
- 95% of mobile web access today is from iPhone
- ~25% of WinMo & BB users download apps
- ~95% of iPhone users download apps
- 100m app downloads in 60 days

The ToolChain

- Interface Builder
- XCode
- iPhone Simulator
- Apple Program Portal
- Devices

Interface Builder

- Powerful, but a little subtle
- Generates objects, not code
 - Safer, more version proof
 - Needs connection to code (outlets, actions)

XCode

- Poor keyboard navigation
- Not as robust or as fast as Visual Studio
- Excellent integration with simulator and devices

Fav XCode shortcuts

Shortcut	Action
⌘ ⌥ ↑	Swap header/code file
⌘ ⌥ ← →	Back forward
⇧ ⌘ E	Full edit
⌘ B	Build
⌘ R	Run
⇧ ⌘ Y	Debugger
^ ⌥ ⌘ R	Clear console
⌘ O	Editor

More XCode Config

- Change XCode prefs Layout to All-in-One (all projects must be closed to change)
- Also, in Mac keyboard prefs->Keyboard Shortcuts, enable tabs to move between "All controls"

Magic: basic emacs keys anywhere on your mac

Objective C

- C with some keywords to add classes (i.e., @interface and @class)
- Lots of square brackets.
- You do not call a method. Instead you send a message to an object (i.e., dynamic runtime)
- No memory management (reference counting)

Objective C Methods

C# / Java

- `objInstance.methodName(param1, param2);`

Objective-C (named parameters)

- `[objInstance methodName:param1
paramName:param2];`

Memory management

Reference counting model

- alloc sets count to 1.
- retain increases count, release decreases.
- autorelease adds object to the autorelease pool
- Usually lasts the duration of the event being processed.

Properties

Syntactic sugar that enables "dot property" syntax.

- Use @property in header file.

- `foo.bar = 10;` OR `[foo setBar:10];`

- `int var = foo.bar;` OR `int var = [foo
bar];`

Other Objective-C Wierdness

- Follow protocols
(i.e., implement interfaces)
- Categories
(named functionality group)

iPhone Simulator

Remarkably consistent with device, but:

- No accelerometer
- Performance better
- Access to some libraries not on device
- Access to file system limited on device

Apple Program Portal

- \$100 / yr to develop or ship to device
- Process works, but remarkably complicated
- Getting apps approved and in AppStore currently takes ~1 week

Device Development

- Need to be in developer program
- Need to get code signing just right
- iPod Touch and iPhone same except for camera and radio

Resources

- <http://developer.apple.com/iphone>
Need an ADC account and have to accept NDA
- http://developer.apple.com/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/chapter_1_section_1.html
- <http://www.stepwise.com/Articles/Technical/2001-03-11.01.html>
The article on mac reference counting
- www.kpcb.com/initiatives/ifund/index.html