Why iPhone Changes Everything
... and how

users
impact $$$

Ben Bederson
Human-Computer Interaction Lab
University of Maryland
## The Platform

<table>
<thead>
<tr>
<th></th>
<th>iPhone</th>
<th>others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large 320x480 screen</td>
<td>Smaller or lower-res screens</td>
<td></td>
</tr>
<tr>
<td>600 MHz CPU</td>
<td>Max 400 MHz</td>
<td></td>
</tr>
<tr>
<td>GPU w/ OpenGL</td>
<td>None (mostly)</td>
<td></td>
</tr>
<tr>
<td>Accelerometer</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><em>High</em> quality touch screen</td>
<td>Pathetic wrongness</td>
<td></td>
</tr>
<tr>
<td>WiFi</td>
<td>mix</td>
<td></td>
</tr>
<tr>
<td>Real web</td>
<td>mix</td>
<td></td>
</tr>
<tr>
<td>The Platform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iPhone</th>
<th>others</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppStore</td>
<td>coming</td>
</tr>
<tr>
<td>They own music</td>
<td>wannabes</td>
</tr>
<tr>
<td>Consistent platforms</td>
<td>Wide variety</td>
</tr>
<tr>
<td>Soup to nuts</td>
<td>Ecosystem of software &amp; hardware</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Platform</th>
<th>Language</th>
<th># Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple iPhone</td>
<td>Objective C</td>
<td>~10 million</td>
</tr>
<tr>
<td>Microsoft WinMo</td>
<td>C# (.NET)</td>
<td>~20 million</td>
</tr>
<tr>
<td>RIM BlackBerry</td>
<td>Java</td>
<td>~40 million</td>
</tr>
<tr>
<td>Nokia Series 60 (Symbian)</td>
<td>C++</td>
<td>???</td>
</tr>
<tr>
<td>Google Android</td>
<td>Java</td>
<td>0 (wait a week)</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>⌘⌥ ↑</td>
<td>Swap header/code file</td>
</tr>
<tr>
<td>⌘⌥ ←→</td>
<td>Back forward</td>
</tr>
<tr>
<td>⇧⌘E</td>
<td>Full edit</td>
</tr>
<tr>
<td>⌘B</td>
<td>Build</td>
</tr>
<tr>
<td>⌘R</td>
<td>Run</td>
</tr>
<tr>
<td>⌘Y</td>
<td>Debugger</td>
</tr>
<tr>
<td>^⌥⌘R</td>
<td>Clear console</td>
</tr>
<tr>
<td>⌘0</td>
<td>Editor</td>
</tr>
</tbody>
</table>
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

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

Objective-C (named parameters)

```objc
[objInstance sendMessage: 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 Weirdness

- Follow protocols (i.e., implement interfaces)
- Categories (named functionality group)
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://www.stepwise.com/Articles/Technical/2001-03-11.01.html
The article on mac reference counting

www.kpcb.com/initiatives/ifund/index.html