iPhone Dev Seminar

for CS 50
Vivek Sant
December 7, 2008

Overview

1. Getting started
2. Resources
3. Types of apps
4. Anatomy of an App
5. Objective-C
6. Hello World
7. A Calculator

1. Getting Started

Getting Started

- Must have OS X Leopard
- Sign up for free Apple Developer account
- Download SDK (http://developer.apple.com/iphone/)
- Start messing around with examples!

2. Resources

Resources

- Apple
  - http://developer.apple.com/iphone/
  - https://developer.apple.com/webapps/
- Stanford's iPhone class
  - http://www.stanford.edu/class/cs193p/
- Google (it's getting better)

3. Types of Apps

- Web app (website)...business as usual...with a few adaptations
- Traditional “iPhone app” - coding in Objective-C

Web apps

1. Page width = 320px

Can limit this during dev:

```
<meta name="viewport"
  content="width=320" />
```

Or for even more control,

```
<meta name="viewport"
  content="width=320;
    initial-scale=1.0;
    maximum-scale=1.0;
    user-scalable=0;" />
```

Some tips from http://www.surveygizmo.com/survey-blog/10-tips-for-developing-iphone-applications/

Web apps

2. Page height = 356px

You can gain 60px by hiding the address bar with:

```
<script type="application/x-
javascript">
  function hideAddressBar()
  { window.scrollTo(0,1); }
</script>
```

But generally don't worry about height

Web apps

3. Big form controls!

Safari on iPhone: can specify w,h for radio and checkbox

Web apps

4. Nifty animation

- Fit with iPhone's theme

- Too much js = long load time;
caution: body onload may be sluggish

Resources:

- Yahoo UI Library: http://developer.yahoo.com/yui/animation/
- Scriptaculous - http://script.aculo.us/
- and of course, developer.apple.com

Web apps

5. `finger != mouse`

Remember this.

So no “onmouseover” and hover events etc!

“Regular” Apps

“Regular” Apps

- Actually use the SDK
- Objective-C
- Steeper learning curve but more power

So where are we?

1. Getting started
2. Resources
3. Types of apps
4. Anatomy of an App
5. Objective-C
6. Hello World
7. A Calculator

4. Anatomy of an App

- Compiled code
  - Your code
  - Cocoa Touch framework code
- Nib files
  - Contains user interface elements (along with other objects)
  - Includes details about object relationships
- Resources (sounds, images, etc)
- Info.plist file (application configuration)

MVC

Model, View, Controller Divides an application into 3 main functional pieces

Model

- Manages the app data and state, not concerned with UI or presentation
- Often persists somewhere

View

- Displays the model objects to the user
- Allows user to manipulate data by responding to events

Controller

- Coordinates the model and the view, keeps the view updated when model changes, etc. Typically where app "logic" is.

5. Objective-C

- Objective-C - strict superset of C;
- .h = header file; .m = implementation file;
- Some syntax:
- Say you declare Person * voter;
- voter.castBallot(); ==> [voter castBallot];
- voter.setAge(21); ==> [voter setAge:21];
- [voter registerForState:@"CA" party:@"Independent"];

5. Objective-C

- With Obj-C 2.0, dot syntax at least for accessor methods:
- Instead of [person setHeight:newHeight];
- can do person.height = newHeight;

- Bool typedef, YES and NO;

- NSObject -- root class; implements mem. mgmt, introspection, object equality

5. Objective-C

- NSString * -- used instead of char *
- In C, constant strings are
  - "text";
- In ObjC, constant strings are
  - @"text"
- Constant strings are NSString instances:
- NSString *aString = @"Hello World!";

- NSMutableString -- allows string to be modified

Let's jump in --

6. Hello World!