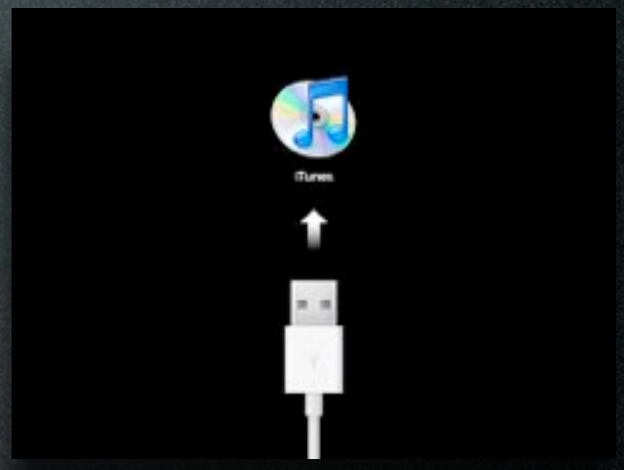
Development of Computer Graphics and Digital Image Processing on the iPhone

Luciano Fagundes
(luciano@babs2go.com.br)
Rafael Santos
(rafael.santos@lac.inpe.br)

babs2go

Motivation

iOS Devices



Dev Basics

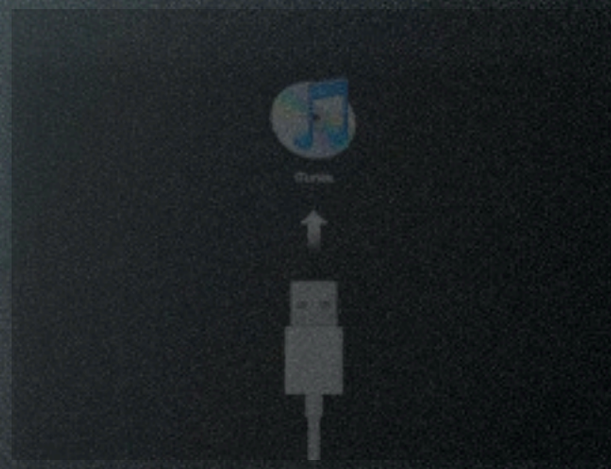
From Concept to the App Store

Agenda

babs2go

Motivation

iOS Devices



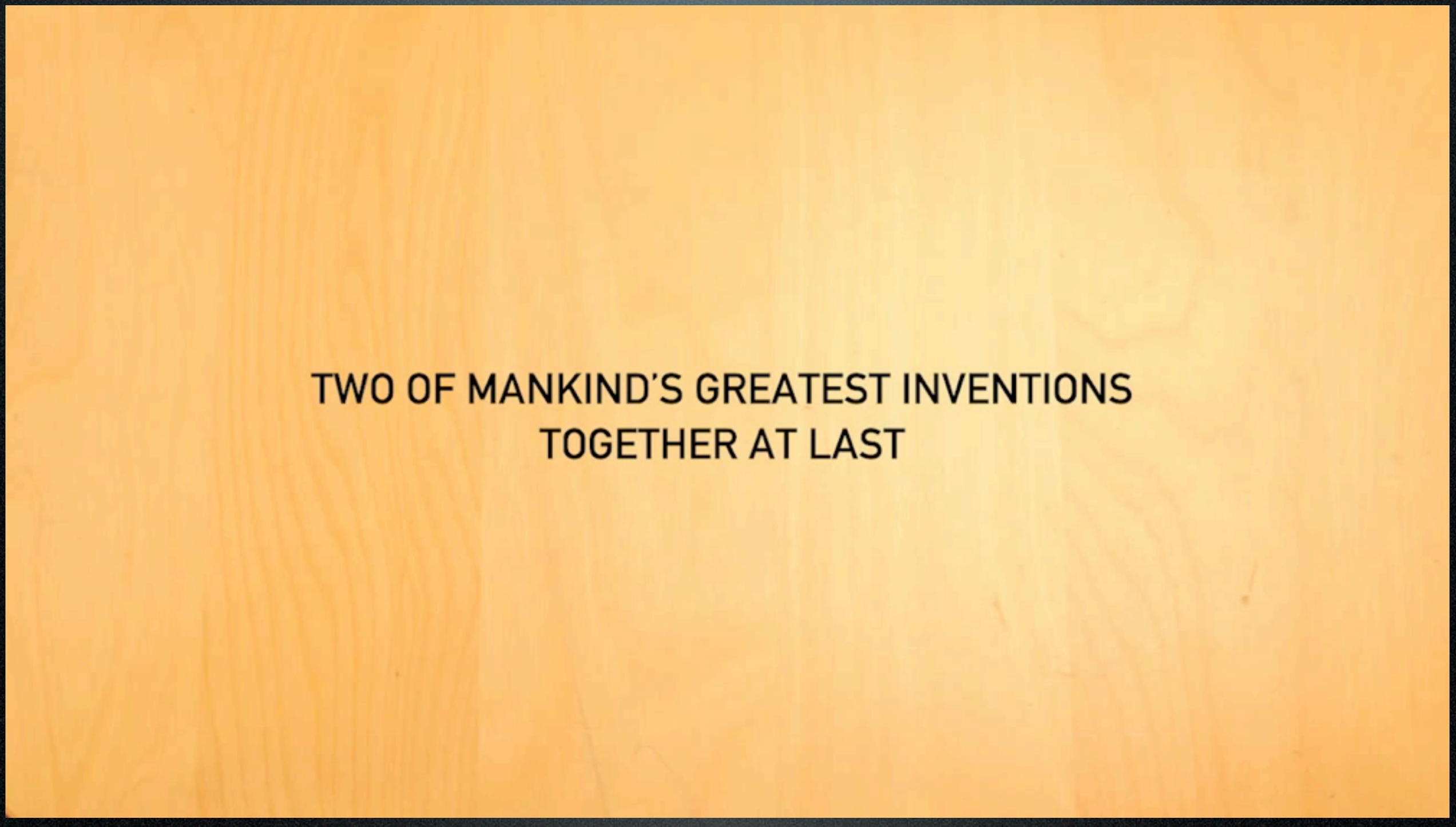
Dev Basics

From Concept to the App Store

Agenda

Motivation

Motivation



Motivation

- Multi-purpouse device
- Wide Community (Millions of iOS devices world-wide)
- Easy to Deploy !!! (App Store)
- Apps can and are changing the World
- (Size of the Market) See Gigaom Infograph

Motivation

Old WAP-based Mobile Phones

iPhone

iPad

iPod Touch

iOS Devices

iPhone

iPad

iPod Touch

Camera +
Phone +
WiFi + 3G

iOS Devices

babs2go

iPhone

Camera +
Phone +
WiFi + 3G

iPad

WiFi + 3G

iPod Touch

iOS Devices

babs2go

iPhone

Camera +
Phone +
WiFi + 3G

iPad

WiFi + 3G

iPod Touch

WiFi

iOS Devices

Tech Benchmark

iPhone

	iPhone 4	HTC EVO 4G	Nokia N8	Palm Pre Plus	HTC HD2
					
Platform	iOS 4	Android 2.1 with Sense	Symbian^3	webOS	Windows Mobile 6.5
Processor	Apple A4	1GHz Qualcomm Snapdragon	680MHz ARM11-based	600MHz TI OMAP3430	1GHz Qualcomm Snapdragon
Storage	16GB / 32GB internal	440MB internal, microSDHC expansion	16GB internal, microSDHC expansion	16GB	Approx. 200MB internal, microSDHC expansion
Cellular	Quadband GSM, pentaband HSPA	CDMA, EV-DO Rev. A, WiMAX	Quadband GSM, pentaband HSPA	CDMA / EV-DO Rev. A or quadband GSM / dualband HSPA	Quadband GSM, dualband HSPA
WiFi	802.11b/g/n	802.11b/g	802.11b/g/n	802.11b/g	802.11b/g[1]
Display size	3.5 inches	4.3 inches	3.5 inches	3.1 inches	4.3 inches
Display resolution	960 x 640	800 x 480	640 x 360	480 x 320	800 x 480
Display technology	IPS LCD	LCD	AMOLED	LCD	LCD
Integrated TV-out	No	HDMI	HDMI	No	No
Primary camera	5 megapixel AF, LED flash	8 megapixel AF, LED flash	12 megapixel AF, xenon flash	3 megapixel, LED flash	5 megapixel AF, LED flash
Secondary camera	VGA	1.3 megapixel	VGA	None	None
Video recording	720p at 30fps	720p at 24fps	720p at 25fps	VGA at 30fps	VGA at 30fps
Video calling	Yes (WiFi only)	Yes	Yes	No	No
Location / orientation sensors	AGPS, compass, accelerometer, gyroscope	AGPS, compass, accelerometer	AGPS, compass, accelerometer	AGPS, accelerometer	AGPS, compass, accelerometer
SIM standard	Micro SIM	N/A	SIM	SIM (on GSM variant)	SIM
Quoted max talk time	7 hours on 3G, 14 hours on 2G	6 hours	5.83 hours on 3G, 12 hours on 2G	5.5 hours on Verizon, 5 hours on AT&T	5.33 hours on 3G, 6.33 hours on 2G
Quoted max media playback time	40 hours audio, 10 hours video	None quoted	50 hours audio, 6 hours video	None quoted	12 hours audio, 8 hours video
Weight	137 grams / 4.8 oz.	170 grams / 6.00 oz.	135 grams / 4.76 oz.	135 grams / 4.76 oz.	157 grams / 5.54 oz.
Dimensions	115.2 x 58.6 x 9.3mm	122 x 66 x 13mm	113.5 x 59 x 12.9mm	100.5 x 59.5 x 16.95mm	120.5 x 67 x 11mm

Engadget

Tech Benchmark

iPhone4/iPad

“The iPad's graphics capabilities come from a PowerVR SGX GPU, similar to the one found in the iPhone 3GS and iPod Touch. If the iPad is using the same PowerVR SGX 535 as in the 3GS, it can render about **28 million polygons/ second** ... “

PCMagazine

IPhone 3GS vs 3G

IPhone 3GS vs 3G

Mad Monkey TECH

3G vs 3GS Speed Test

- 36K Polys
- 7 Textures 512x512
- 14 Textures 256x256
- 4 Textures 128x128

iPhone4 vs 3Gs

"Early benchmarks of the iPhone 4 show it to be 31% faster than the iPhone 3GS according to the Geekbench 2 app...."

Macrumors

iPhone4 vs 3Gs

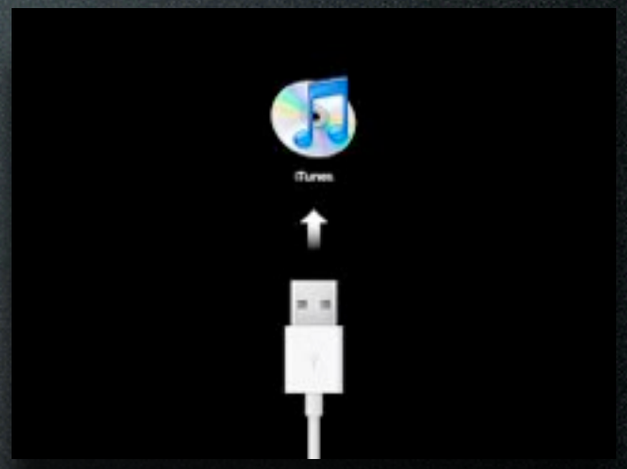
"Early benchmarks of the iPhone 4 show it to be 31% faster than the iPhone 3GS according to the Geekbench 2 app...."

Macrumors

Brings iPhone4 performance to ~36 Million Polygons/sec

Motivation

iOS Devices



Dev Basics

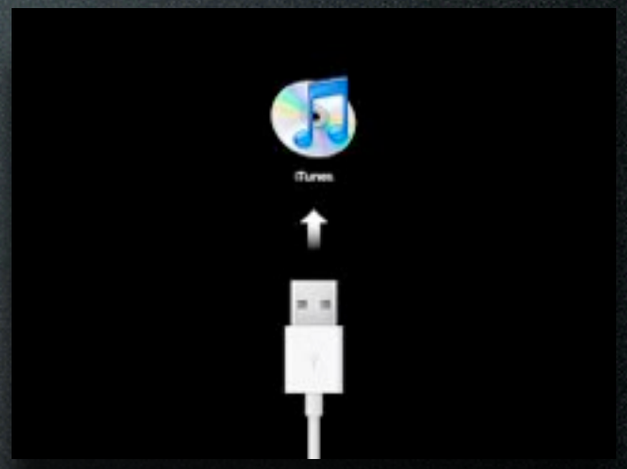
From Concept to the App Store

Agenda

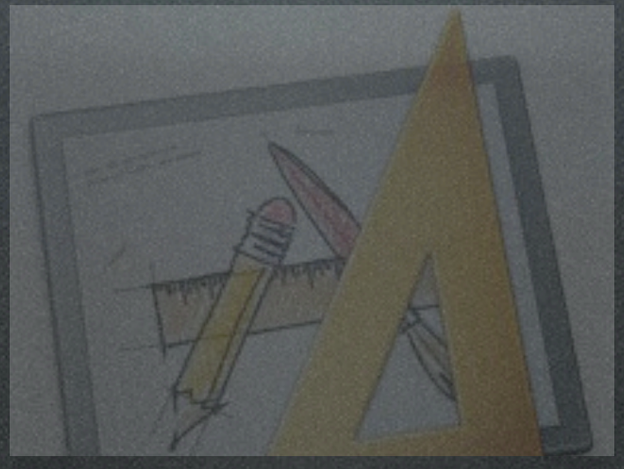
babs2go

Motivation

iOS Devices



Dev Basics



From Concept to the App Store

Agenda

What would you need to start Building Native Apps??

Dev Basics

What would you need to start Building Native Apps??

MacOS Device

Dev Basics

What would you need to start Building Native Apps??

MacOS Device

iPhone
Dev Center

Dev Basics

What would you need to start Building Native Apps??

MacOS Device

iPhone
Dev Center

XCode
+ iOS SDK

Dev Basics

Dev Basics

- Objective C
  - It is a superclass of ANSI C
  - It brings Object Orientation to traditional ANSI C
  - It is NOT C++
  - Objective C++ works with C++

Dev Basics

- Objective C requires Header Files (*.h) and Implementation Files (*.m)

- Objective C++ requires Header Files (*.h) and Implementation Files (*.mm)

Dev Basics

- A header file from HelloWorld Project:

```
//
//  HelloWorldViewController.h
//  HelloWorld
//
//  Created by Luciano Fagundes on 19/08/10.
//  Copyright Pessoal 2010. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface HelloWorldViewController : UIViewController {

}
```

Dev Basics

```
#import
"HelloWorldViewController.h"

@implementation
HelloWorldViewController
- (id)initWithNibName:(NSString
*)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil {
    return self;
}

- (void)loadView {
}

- (void)viewDidLoad {
    [super viewDidLoad];
}
```

```
- (void)
didReceiveMemoryWarning {

    [super
didReceiveMemoryWarning];
}

- (void)viewDidUnload {

}


- (void)dealloc {
    [super dealloc];
}

@end
```

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundleOrNil {
    return self;
}
```

Object Method in Objective C

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundle {

    return self;
}
```

In Java it would be :

```
public (Object*)initWithNibNameBundle(String
nibName, Bundle nibBundle)
{
    return self
}
```

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundle {

    return self;
}
```

In Java it would be :

```
public (Object*)initWithNibNameBundle(String
nibName, Bundle nibBundle)
{
    return self
}
```

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundle {

    return self;
}
```

In Java it would be :

```
public (Object*)initWithNibNameBundle(String
nibName, Bundle nibBundle)
{
    return self
}
```

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundle {

    return self;
}
```

In Java it would be :

```
public (Object*)initWithNibNameBundle(String
nibName, Bundle nibBundle)
{
    return self
}
```

Dev Basics

```
-(id)initWithNibName:(NSString *)nibName
              bundle:(NSBundle *)nibBundle {

    return self;
}
```

In Java it would be :

```
public (Object*)initWithNibNameBundle(String
nibName, Bundle nibBundle)
{
    return self
}
```

Dev Basics

Sending a Message to an Object

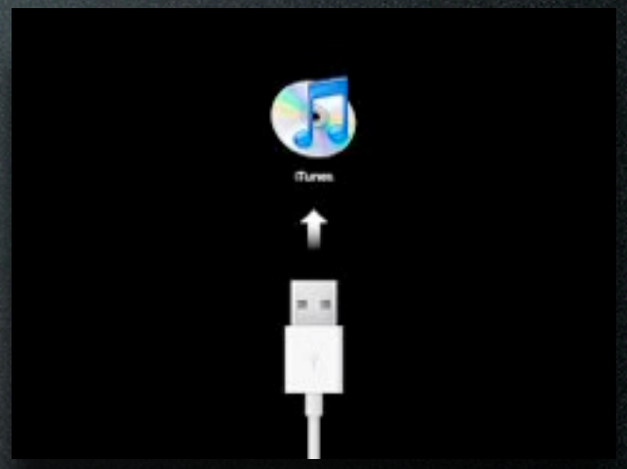
```
[receiver doSomething]
[receiver doThis:this andDoThat:that]
```

In Java it would be :

```
receiver.doSomething()
receiver.DoThisAndDoThat(this,that)
```

Motivation

iOS Devices



Dev Basics

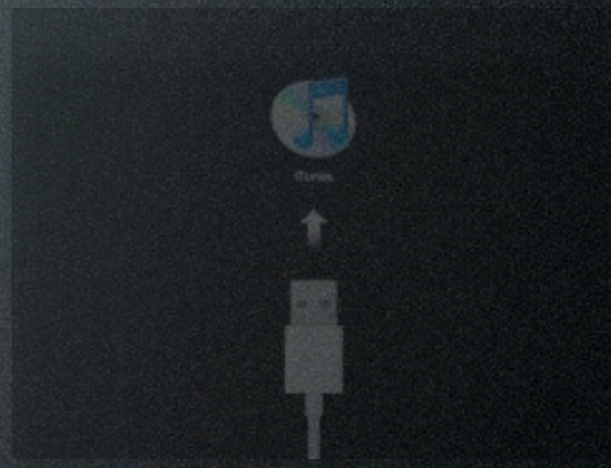
From Concept to the App Store

Agenda

babs2go

Motivation

iOS Devices



Dev Basics

From Concept to the App Store

Agenda

Demo 1: XCode Interface Builder

Show the entire environment:

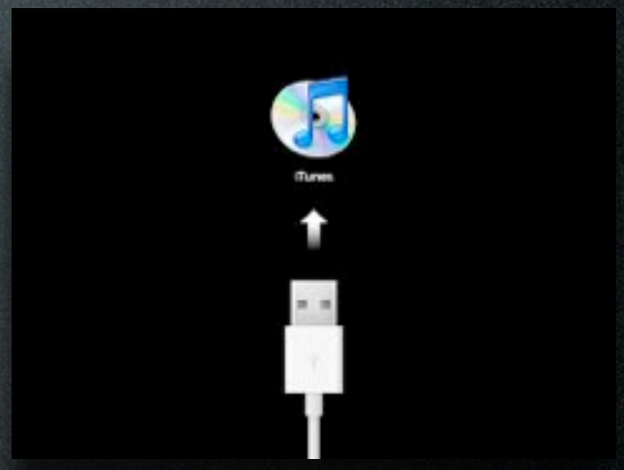
-Create New Project
-Place a Button at the Screen
-Create Action and Outlet

- Show iPhone and iPad Simulator
- Load App into iPhone Device
- App should have a label and a button;
- Pressing a button indicates that text show Hello World

babs2go

Motivation

iOS Devices



Dev Basics

From Concept to the App Store

Agenda

App Store


iTunes
ENTERTAINMENT
TOP PAID APPS
TOP FREE APPS
Page 1 of 873
Sort by: Release Date


VIVO
17:56
Categories
Entertainment
Top Paid
Top Free
Release Date
Babs2Go
PACMAN Reloaded
$0.99
9 Ratings
HuddleTech
320 Wallpapers
$0.99
No Ratings
Marcelo Queiroz
Damas
$0.99
18 Ratings
NextApps Inc.
Boooooly!!
$0.99
5 Ratings
DreamWiz Inc.
Featured
Categories
Top 25
Search
Updates

App Store

Create an Account on the “IPhone Developer Program”

Build you App

- Download XCode
- Develop / Test
- Download your Digital Certificate
- Sign your App

Distribution

- Access iTune Connect
- Enter all the marketing material for your App, including : Multiple Languages, price, description, snapshots, target countries, etc
- Submit App to Apple for Approval

App Store

Maintenance

- Monitor your Sales Reports from the entire World cia iTunes Connect (https://itunesconnect.apple.com/WebObjects/iTunesConnect.woa)
- Monitor your App reviews (Customer feedback) for the several different countries
- Fix Bugs / Evolve the App
- Submit updated App to Apple and start all over

GET RICH !!

- You will get 70% of the price you have set for your App
- Another way would be launching free Apps monetized by Advertisement (iAd, Admob and others)

App Store

Advantages:

- World Wide Exposure
- Very Hot Market for independent developers
- Very quick product cycle
- Up to the second user feedback

App Store

Risks:

- Brutal Competition (Fast and Flexible)
- Apps can fade among the others
- Consumer has total control of the market; very easy to discard and provide a bad review for your App

Image Processing With Quartz

Showing Images

```
UIImageView * imgView =
        [[UIImageView alloc] initWithFrame(10,10,100,100)]];
UIImage * img =
        [[UIImage alloc] initWithURL: <<URL for the image>>];
imgView.image = img;
[img release];
```

Acquiring Images

```
UIImagePickerController *ipc =
        [[UIImagePickerController alloc] init];

// Get images from the photo library.
ipc.delegate = self;
ipc.sourceType =
        UIImagePickerControllerSourceTypePhotoLibrary;

[self presentModalViewController:ipc animated:YES];
```

Receiving Image

```
-(void) imagePickerController:
                        (UIImagePickerController *)picker
didFinishPickingMediaWithInfo:
                        (NSDictionary *)info
{
    [self displayImage:[info objectForKey:
                        @"UIImagePickerControllerOriginalImage"]];
    [self dismissModalViewControllerAnimated:
                        YES];
    [picker release];
}
```

Receiving Image

```
-(void) imagePickerController:
                        (UIImagePickerController *)picker
didFinishPickingMediaWithInfo:
                        (NSDictionary *)info
{
    [self displayImage:[info objectForKey:
                        @"UIImagePickerControllerOriginalImage"]];
    [self dismissModalViewControllerAnimated:
                        YES];
    [picker release];
}
-(void) imagePickerControllerDidCancel:
                        (UIImagePickerController *)picker;
```

Drawing

```
-(UIImage *)drawPieChart {
//Define an area where the chart will be
drawn

CGRect workArea = CGRectMake
(0,0,self.view.frame.size.width,self.vie
w.frame.size.height);

// set the center of the pie chart !
CGPoint chartCenter = CGPointMake
(workArea.size.width/2,
workArea.size.height/2);
```

Drawing

```
//Manually creates a a Graphical Context

int pixelsWide = workArea.size.width;
int pixelsHigh = workArea.size.height;
CGContextRef ctx=NULL;
CGColorSpaceRef colorSpace;
void* bitmapData;
int bitmapByteCount;
int bitmapBytesPerRow;

// find dimension of the image regarding
its bytes
bitmapBytesPerRow =(pixelsWide*4);
```

Drawing

```
//RGBA !!
bitmapByteCount =
(bitmapBytesPerRow*pixelsHigh);

//Allocate image buffer
bitmapData=malloc(bitmapByteCount);
if(bitmapData==NULL)
{
 fprintf (stderr,"Memorynotallocated!");
 return NULL;
}

//create RGB color space
colorSpace=
     CGColorSpaceCreateDeviceRGB();
```

Drawing

```
//Create a Ghraphic Context with the
just created buffer

ctx=CGBitmapContextCreate
(bitmapData,pixelsWide,pixelsHigh,
        8,//bitspercomponent
        tipliedLast);
        bitmapBytesPerRow,
        colorSpace,
        kCGImageAlphaPremultipliedlast);
```

Drawing

```
// Returns NULL
if Context creation failed
if(ctx==NULL)
{  //5
    free(bitmapData);
} //release color space because it is no
 longer needed

CGColorSpaceRelease(colorSpace);
 //Clear drawing area !
CGContextClearRect(ctx, workArea);
```

Drawing

```
// Create a sample array of values !
 NSMutableArray * dataSet =
[[NSMutableArray alloc] init];
   [dataSet addObject:[NSNumber
                numberWithInt:10]];
   [dataSet addObject:[NSNumber
                numberWithInt:15]];
[dataSet addObject:[NSNumber
                numberWithInt:30]];
[dataSet addObject:[NSNumber
                numberWithInt:5]];
[dataSet addObject:[NSNumber
                numberWithInt:18]];
```

Drawing

```
 // Create an array of colors for each
pie slice

NSMutableArray * colors =
 [[NSMutableArray alloc] init];
 [colors addObject:[UIColor blueColor]];
 [colors addObject:[UIColor greenColor]];
 [colors addObject:[UIColor redColor]];
 [colors addObject:[UIColor grayColor]];
 [colors addObject:[UIColor whiteColor]];
```

Drawing

```
// sum all values in the array
float total = 0;
for ( NSNumber * iTmp in dataSet )
{
    total+=[iTmp intValue];
}
// Plots the Pie Chart
float startDegree = 0;
float endDegree = 0;
float radius = workArea.size.width/2;

//Plot Items
int item;
CGContextSetStrokeColorWithColor(ctx,
[UIColor blackColor].CGColor );
```

Drawing

```
for(inti=0;i<dataSet.count;i++) {
  item = [(NSNumber*)[dataSet
                objectAtIndex:i]
           intValue];
 endDegree += (item*360.0f);  CGContextSetFillColorWithColor
     (ctx, ((UIColor*)[colors
             objectAtIndex:i]).CGColor );

 CGContextSetLineWidth(ctx, 5); CGContextMoveToPoint(ctx,
      chartCenter.x, chartCenter.y);
 CGContextAddArc(ctx, chartCenter.x,
       chartCenter.y, radius-15,
       startDegree*M_PI/180.0f,
       endDegree*M_PI/180.0f, 0);
 CGContextFillPath(ctx);
 startDegree = endDegree;
}
```

Drawing

```
// Create image to be returned from the
// Graphical Context

CGImageRef img =
     CGBitmapContextCreateImage(ctx);
UIImage * ret = [UIImage
                    imageWithCGImage:img]
 // Free up all remaining memory
 free(CGBitmapContextGetData(ctx));
 CGContextRelease(ctx);
 CGImageRelease(img);
 return ret;
}
```

Exporting to Photo Library

```
UIImageWriteToSavedPhotosAlbum
        ([imageView image], nil,nil,nil);
```

Case of Study
Demo 2:

Sample CG Apps