

Having fun with Ruby Motion



Who am I?

- Happy husband and father of two little girls
- iOS dev at Sergata Mobile doing Objective-C by day
- @yaakov_g 
- [github/yasha0x80](https://github.com/yasha0x80) 
- <http://www.yaakov.io>



Here is where I play with iOS stuff

The screenshot shows the Sergata website homepage. At the top, there's a navigation bar with links for HOME, ABOUT, SERVICES, PORTFOLIO, CONTACT, TECHNOLOGY, EXIT COMPETITION, and BLOG, along with social media icons for Facebook and Twitter. Below the navigation is a large red banner featuring the LoveMe mobile application icon and the text "LOVEME". It also displays two smartphones and a desktop monitor, all showing the LoveMe interface. A "READ MORE" button is visible. Below the banner, there are tabs for WEB, MOBILE APPS (which is selected), DESKTOP, VIDEO, and ON-SITE. The "MOBILE APPS" section contains a "Web Development" section with text about Sergata's expertise in dynamic websites and web applications. To the right, there's a "FEATURED CUSTOMERS" section with logos for PayPal, uiu, Golan, MATOMY, frimper, BIG BLUE, KENSHOO, Mind Lab, Kontera, and bezeq. At the bottom right of the page is a "Sergata" logo.

sergata.com

Colombia



RubyMotion Installer

Welcome to RubyMotion

RubyMotion has been successfully set up on your system.

Open the Getting Started guide.

[Close the Installer](#)

So *what* is
RubyMotion?

Toolchain for building native apps for Mac OSX and iOS



Ingredients



Some ingredients:

Some ingredients:

- LLVM compiler (LLVM system compiler)

Some ingredients:

- LLVM compiler (LLVM system compiler)
- RubyMotion's own ruby 1.9.2

Some ingredients:

- LLVM compiler (LLVM system compiler)
- RubyMotion's own ruby 1.9.2
- some other stuff like a parser from MRI

Some ingredients:

- LLVM compiler (LLVM system compiler)
- RubyMotion's own ruby 1.9.2
- some other stuff like a parser from MRI
- scripts to set things up

HipByte/RubyMotion

Github, Inc. github.com/HipByte/RubyMotion

This repository Search or type a command Explore Gist Blog Help

PUBLIC HipByte / RubyMotion Watch 54 Star 371 Fork 82

318 commits 4 branches 0 releases

branch: master RubyMotion /

add builtin command for lldb

Watson1978 authored 10 days ago

- lib app.short_version defaults to app.version
- lldb add builtin command for lldb
- test add tests for and fix location specific taps
- .gitignore Test that an actual different storyboard is loaded.
- LICENSE initial revision
- README.rdoc Update README.rdoc
- Rakefile Rake tasks that make development and testing easier.
- ripper18.sh add ripper gem for Ruby18

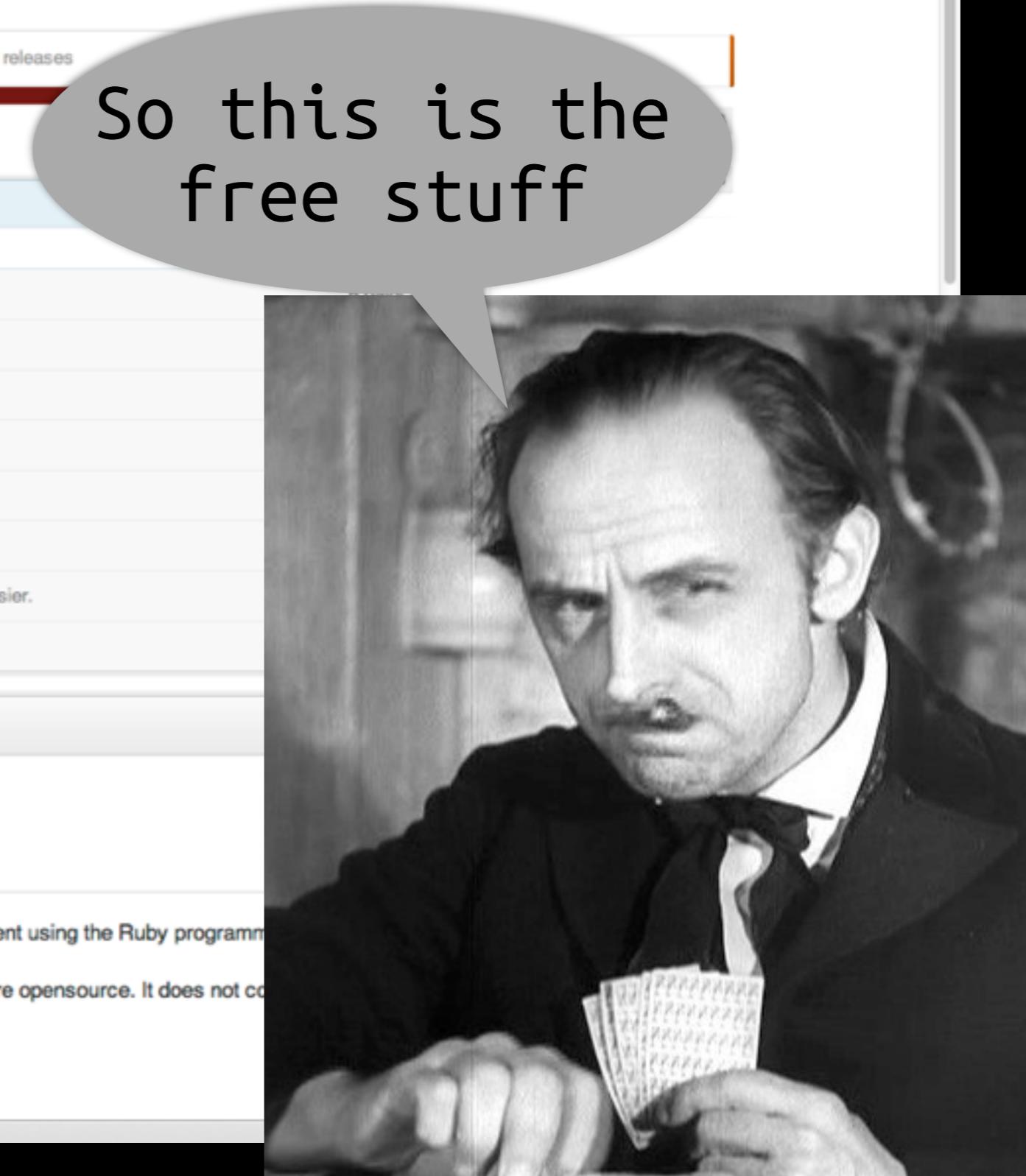
README.rdoc

RubyMotion

RubyMotion is a commercial toolchain for iOS and OS X development using the Ruby programming language.

This repository contains the parts of the RubyMotion product that are open source. It does not contain the full product, which can be purchased at www.rubymotion.com.

Contents



It's not a bridge!



So you can feel at home...



So...

RubyMotion = Ruby in your pocket?

Not exactly...

The Compiler

The Compiler



hello.rb

```
def hello(what)
  puts "Hello {what}"
end
```

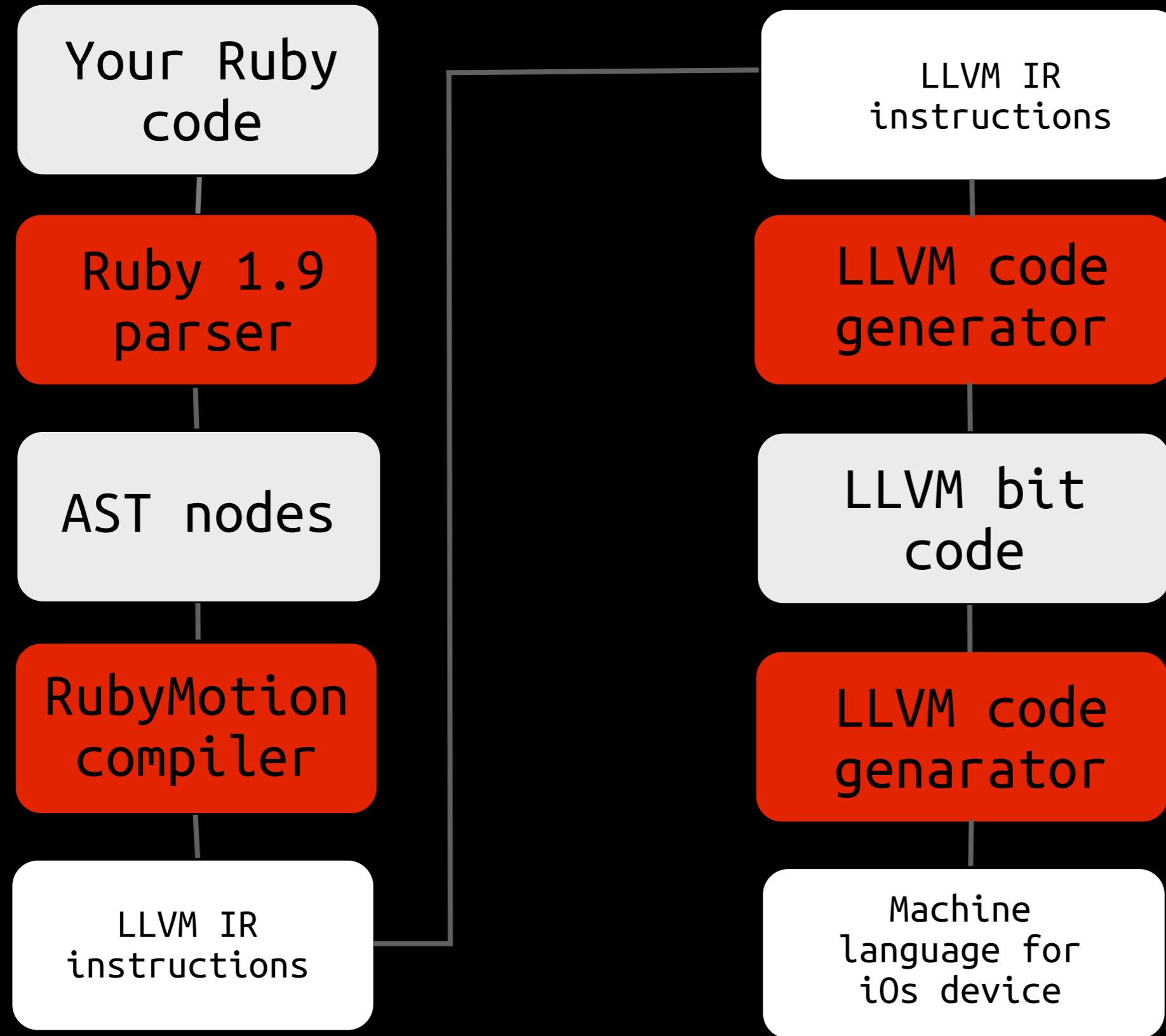
LLVM bitcode

```
define internal i32 @"rb_scope_hello_"
(i32 %self, i8 %sel, i32 %what) {
MainBlock:
etc. .....
%2 = load i8** @0
%3 = call i32 @rb_str_new(i8*getelementptr...)
%4 = load i32* %0
etc. .....
```

LLVM IR assembly

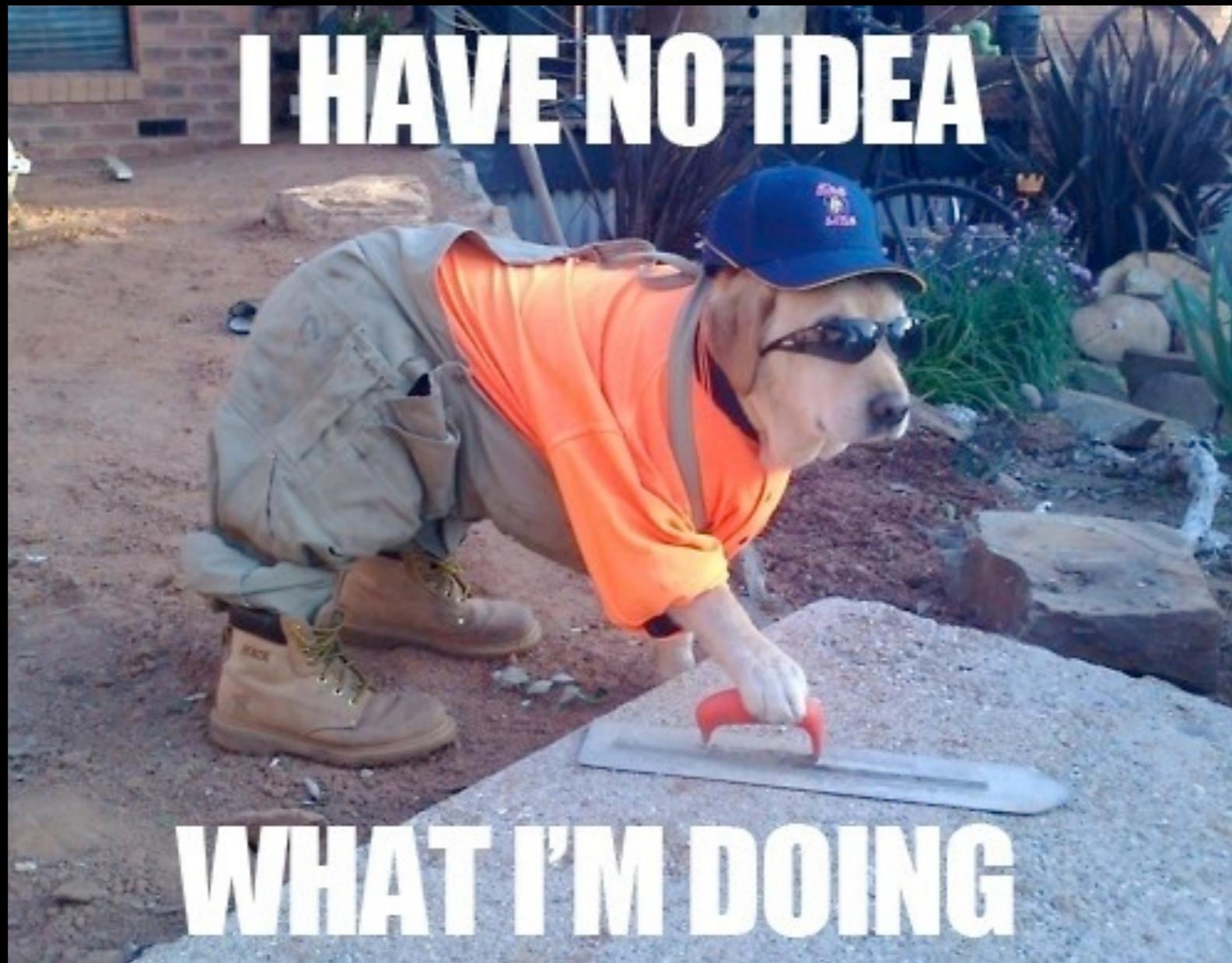
```
.align 44, 0x90
“_rb_scope__hello:_”:
Ltmp732:
Lfunc_begin82:
    .loc 1 0 0
## BB#0:
    pushl %ebp
#etc...
    movl %eax, -28(%ebp)
    leal __unnamed_5-L82$pb(%ebx),%eax
```

#etc...



I HAVE NO IDEA

WHAT I'M DOING



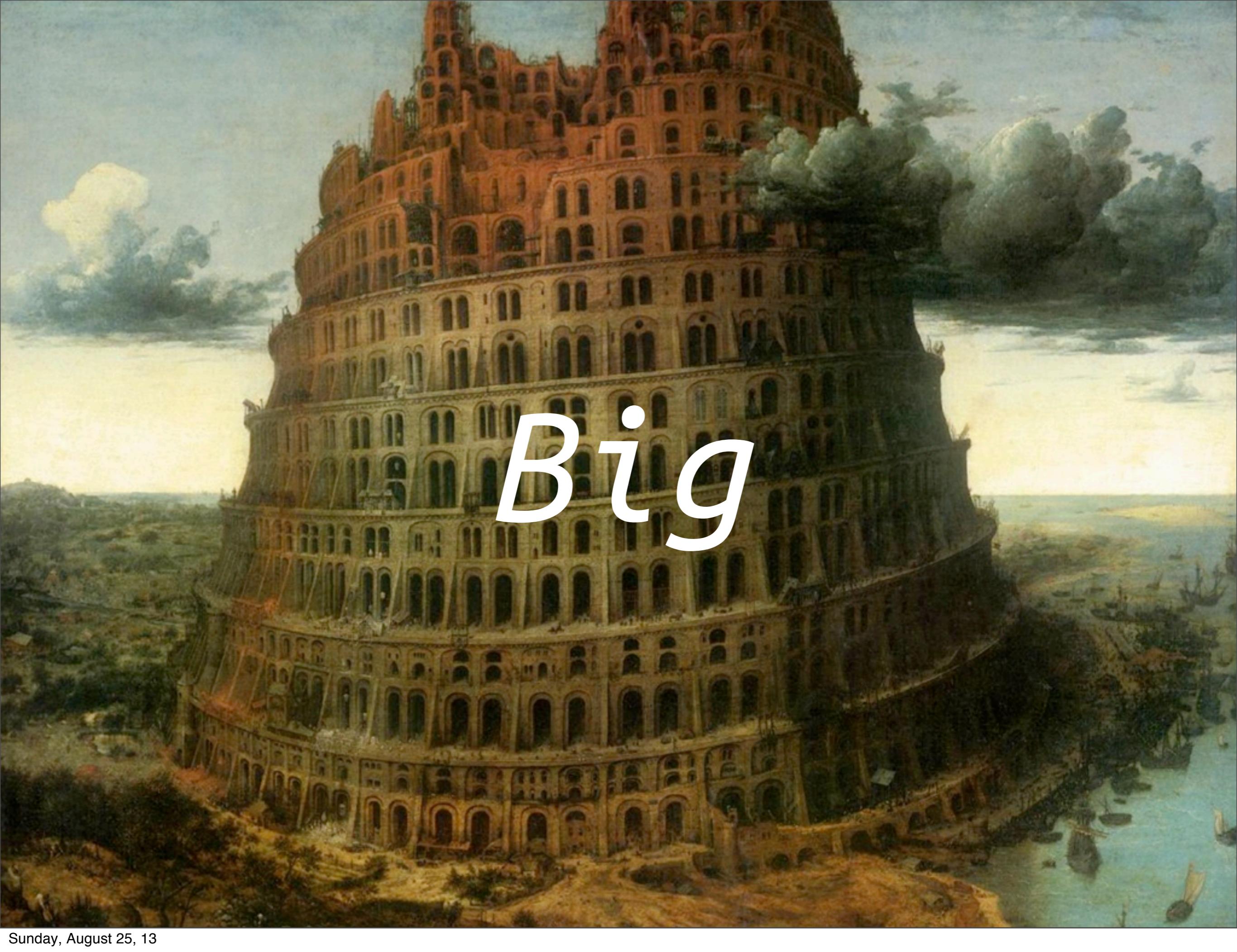
A lot of stuff to talk
about but..

So lets start with
some iOS



Do I need to
learn
Objective-C?

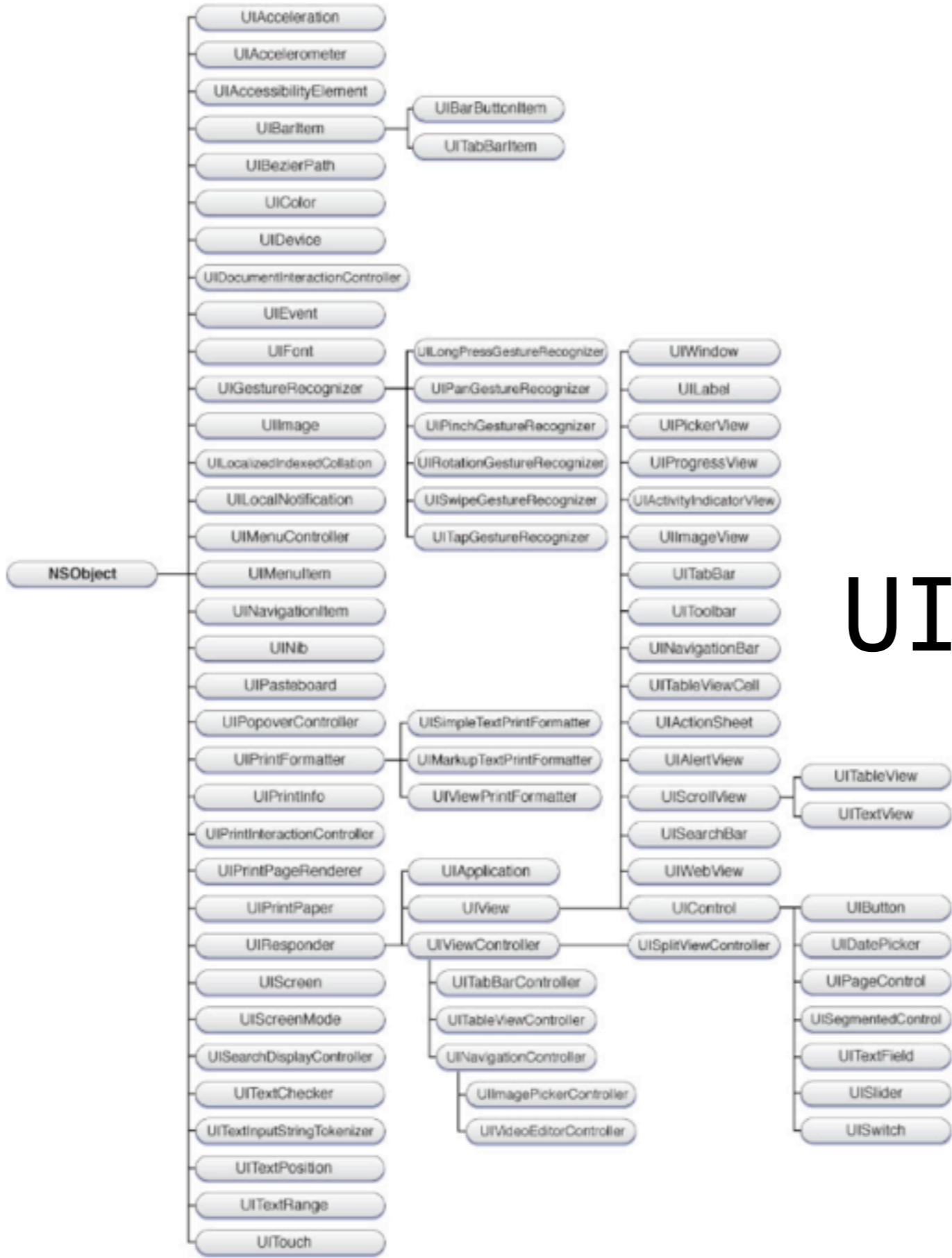
Cocoa and Cocoa-Touch
are . . .



Big

ios }
OS X }

DOCUMENTATION	
▼ iOS 6.1 doc set	
► C Classes	537
► Pr Protocols	132
► C Categories	23
► M Methods	8711
► f Functions	3676
► P Properties	2226
► T Types	1219
► V Constants	7065
► # Macros	446
► G Guides	431
► S Samples	150
▼ OS X 10.8 doc set	
► C Classes	1014
► Pr Protocols	223
► C Categories	73
► M Methods	18546
► B Bindings	548
► f Functions	15796
► P Properties	1623
► T Types	4219
► V Constants	17845
► # Macros	2327
► G Guides	765
► S Samples	364



UIKit framework



{

RubyMotion	
▶ C	Classes 768
▶ Pr	Protocols 120
▶ M	Methods 9953
▶ V	Constants 2016

768 Classes

120 Protocols

9953 Methods

2016 Constants

Apple docs

buttonWithType:

Creates and returns a new button of the specified type.

+ (id)buttonWithType:(UIButtonType)buttonType



Parameters

buttonType

The button type. See [UIButtonType](#) for the possible values.

Return Value

A newly created button.

Discussion

This method is a convenience constructor for creating button objects with specific configurations. If you subclass [UIButton](#), this method does not return an instance of your subclass. If you want to create an instance of a specific subclass, you must alloc/init the button directly.

When creating a custom button—that is a button with the type [UIButtonTypeCustom](#)—the frame of the button is set to (0, 0, 0, 0) initially. Before adding the button to your interface, you should update the frame to a more appropriate value.

Availability

Available in iOS 2.0 and later.

Related Sample Code

[iPhoneMixerEQGraphTest](#)

[Table View Animations and Gestures](#)

[TouchCells](#)

[UICatalog](#)

[XMLPerformance](#)

Declared In

[UIButton.h](#)

RubyMotion docs

Class Method Details

+ (Object) **buttonWithType(buttonType)**

Creates and returns a new button of the specified type. This method is a convenience constructor for creating button objects with specific configurations. If you subclass [UIButton](#), this method does not return an instance of your subclass. If you want to create an instance of a specific subclass, you must alloc/init the button directly. When creating a custom button—that is a button with the type [UIButtonTypeCustom](#)—the frame of the button is set to (0, 0, 0, 0) initially. Before adding the button to your interface, you should update the frame to a more appropriate value.

Parameters:

- **buttonType** ([UIButtonType](#)) — The button type. See [UIButtonType](#) for the possible values.

Returns:

- [\(Object\)](#) — A newly created button.

[W]rappers



THE KLASIX
RAPPERS DELIGHT
VOL. I

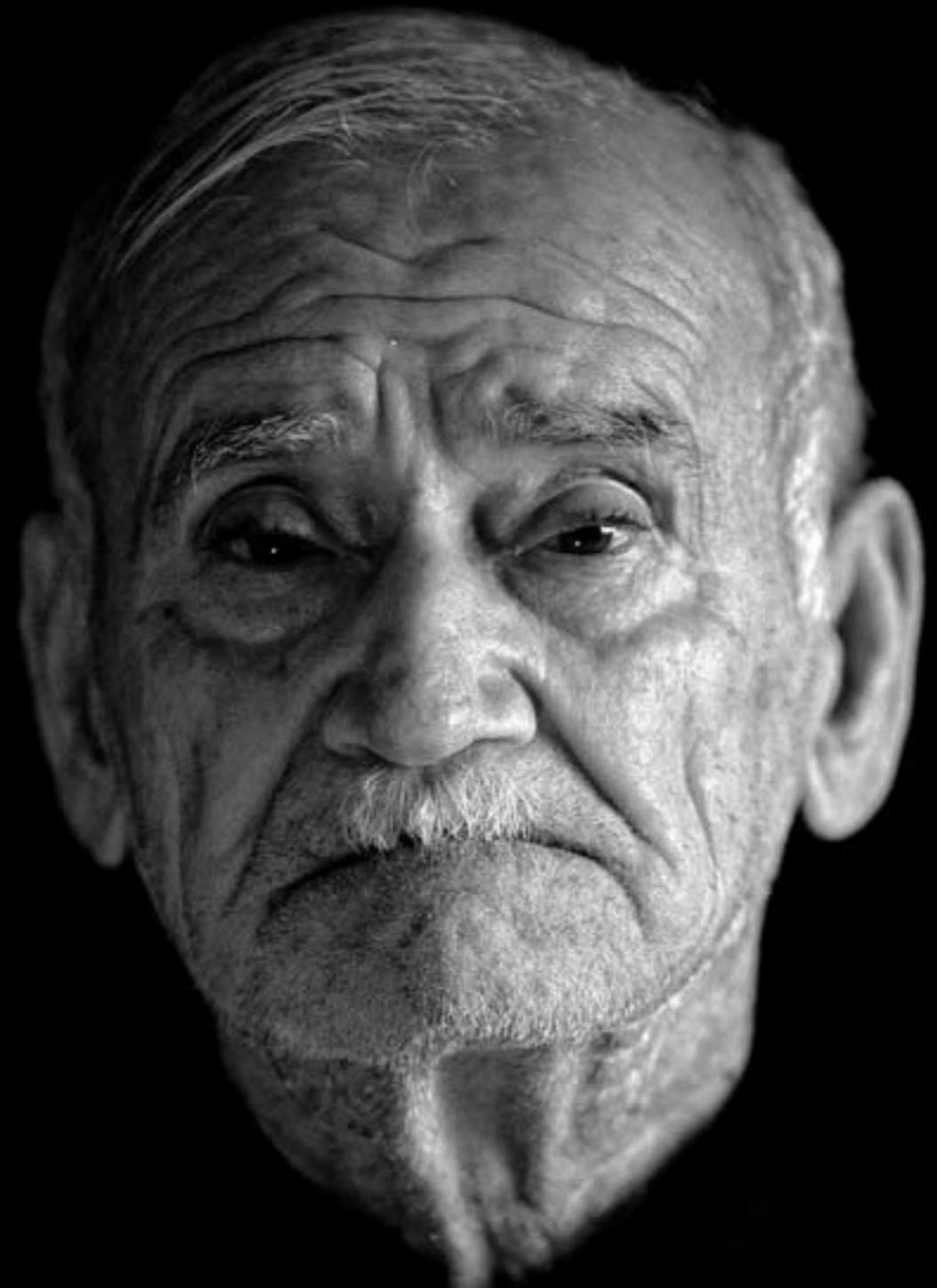
Some syntax



foo.bar(baz)

[foo bar:baz];

The Grandpa



[|]
Smalltalk

[:]

Objective-C class

BigGuy.h

```
#import <Foundation/Foundation.h>

@interface BigGuy : NSObject
@property (nonatomic, strong) NSString *bigGuyName;
- (void)printMessage;
@end
```

BigGuy.m

```
#import "BigGuy.h"

@implementation BigGuy
- (id)init {
    self = [super init];
    if(self) {
        // Do something
    }
    return self;
}
- (void)printMessage {
    NSLog(@"My name is %@", self.bigGuyName);
}
@end
```

A little bit complex



I don't
get it

We know these ones

```
class LittleGuy
attr_accessor :name
def print_guy_name
  puts "My name is #{name}"
end
end
```

Well this looks really
Ruby to me, let's make
some iOS apps



What about this?

```
def glkView(view, drawInRect:rect)
    glClearColor(@curRed, 0.0, 0.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);

    @effect.prepareToDraw

    glBindBuffer(GL_ARRAY_BUFFER, @vertexBuffer)
    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, @indexBuffer)

    glEnableVertexAttribArray(GLKVertexAttribPosition)
    glVertexAttribPointer(GLKVertexAttribPosition, 3, GL_FLOAT, GL_FALSE, 9*4, Pointer.magic_cookie(0))

    glEnableVertexAttribArray(GLKVertexAttribColor)
    glVertexAttribPointer(GLKVertexAttribColor, 4, GL_FLOAT, GL_FALSE, 9*4, Pointer.magic_cookie(3*4))

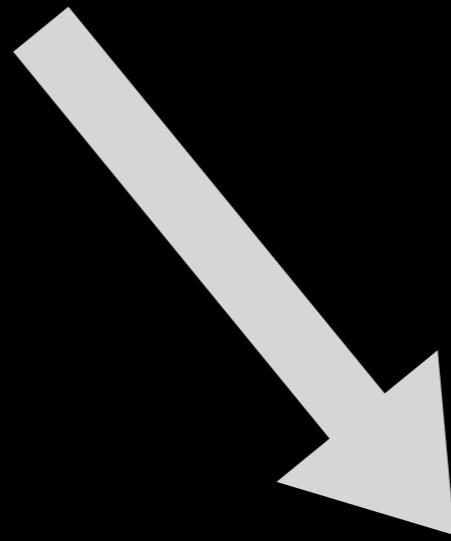
    glEnableVertexAttribArray(GLKVertexAttribTexCoord0);
    glVertexAttribPointer(GLKVertexAttribTexCoord0, 2, GL_FLOAT, GL_FALSE, 9*4, Pointer.magic_cookie(7*4));

    glDrawElements(GL_TRIANGLES, @indices.size, GL_UNSIGNED_BYTE, Pointer.magic_cookie(0))
end
```

just a bit out of place?



???



```
def glkView(view, drawInRect:rect)
glClearColor(@curRed, 0.0, 0.0, 1.0);
glClear(GL_COLOR_BUFFER_BIT);
```

```
def hello(msg,what)
  puts "#{msg} #{what}"
end
```

```
def hello(msg,what:name)
  puts "#{msg} #{name}"
end
```

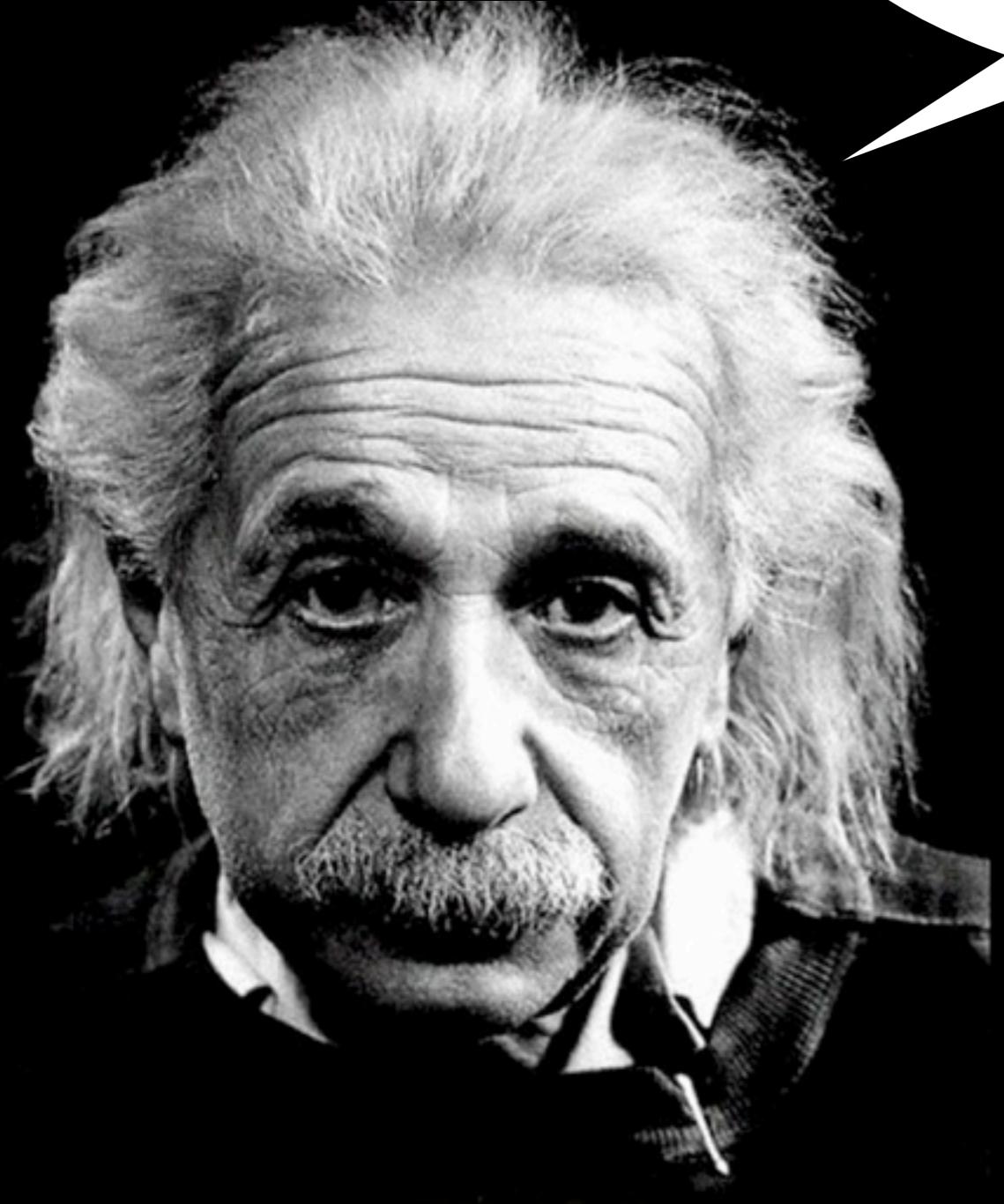
METHOD DEFINITIONS

```
- (void)tableView:(NSTableView *)aTableView  
willDisplayCell:(id)aCell  
forTableColumn:(NSTableColumn *)aTableColumn  
row:(NSInteger)rowIndex
```

that's it?

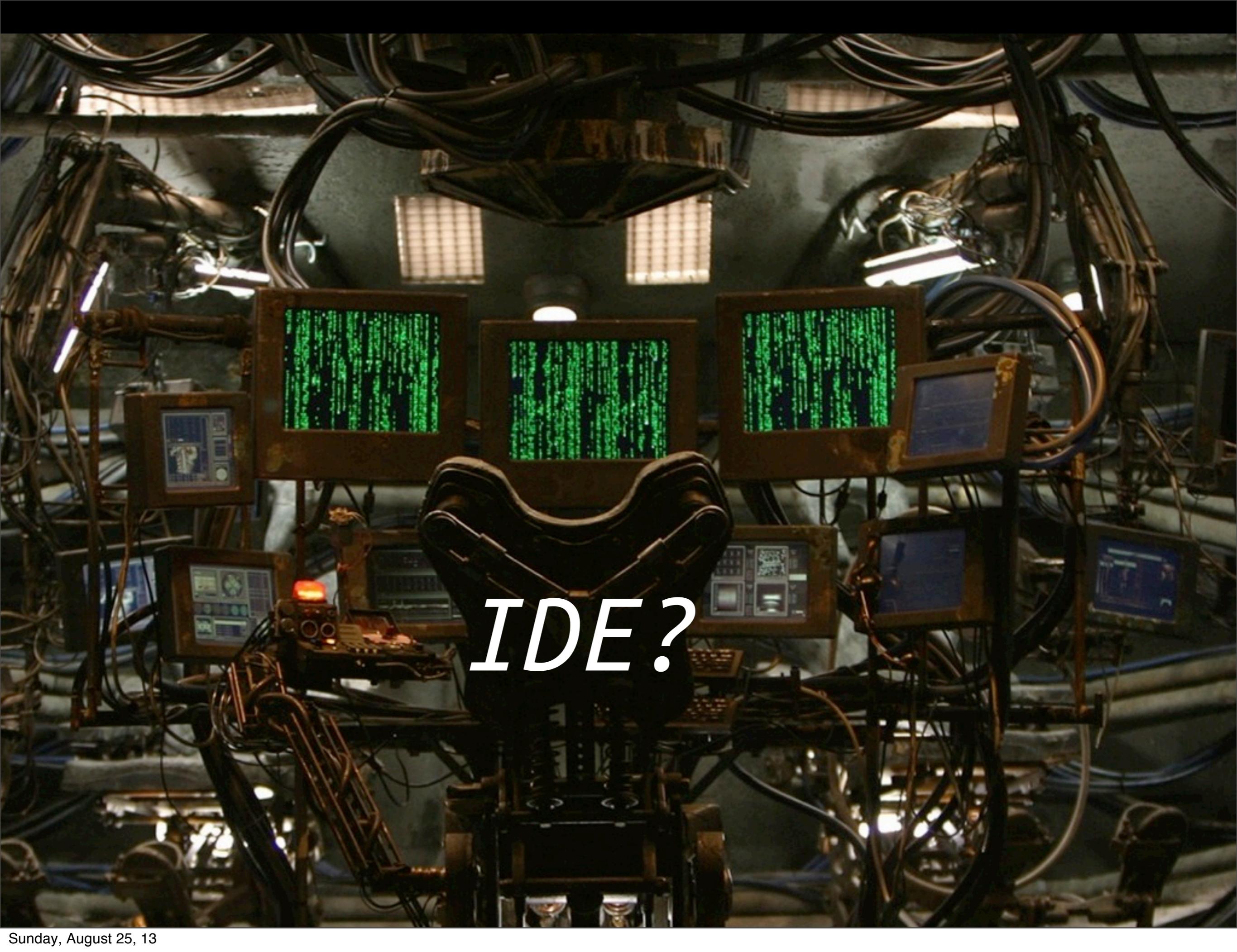


You still need to know
about delegates,
protocols, xibs, blocks,
talking to C code, etc..

A black and white photograph of Albert Einstein's head and shoulders. He has his characteristic wild, grey hair and a full, bushy beard. He is looking slightly to the right of the camera with a thoughtful expression.

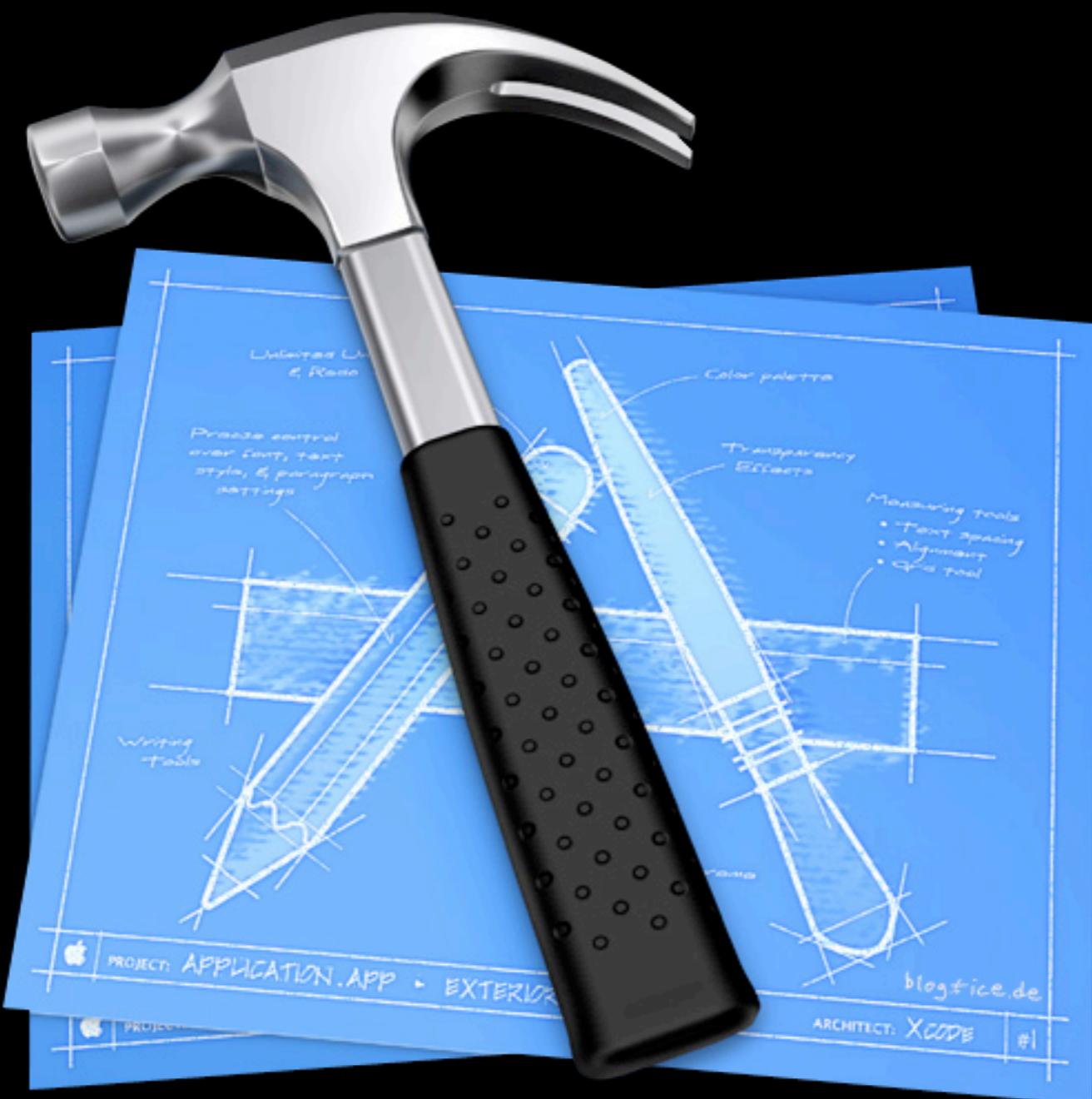
Let's put
something together
to get the idea

but first...
What about the environment?



IDE?

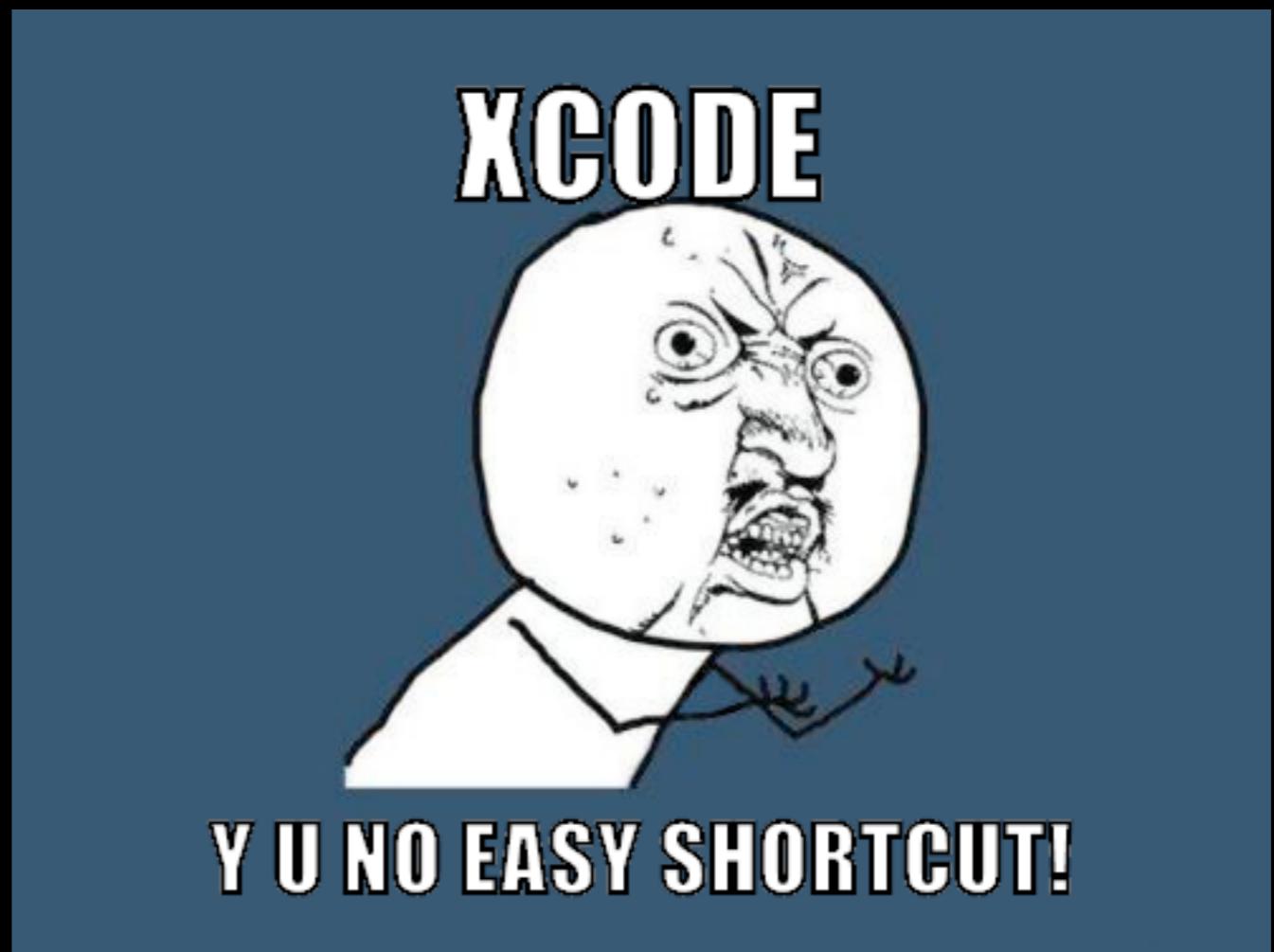
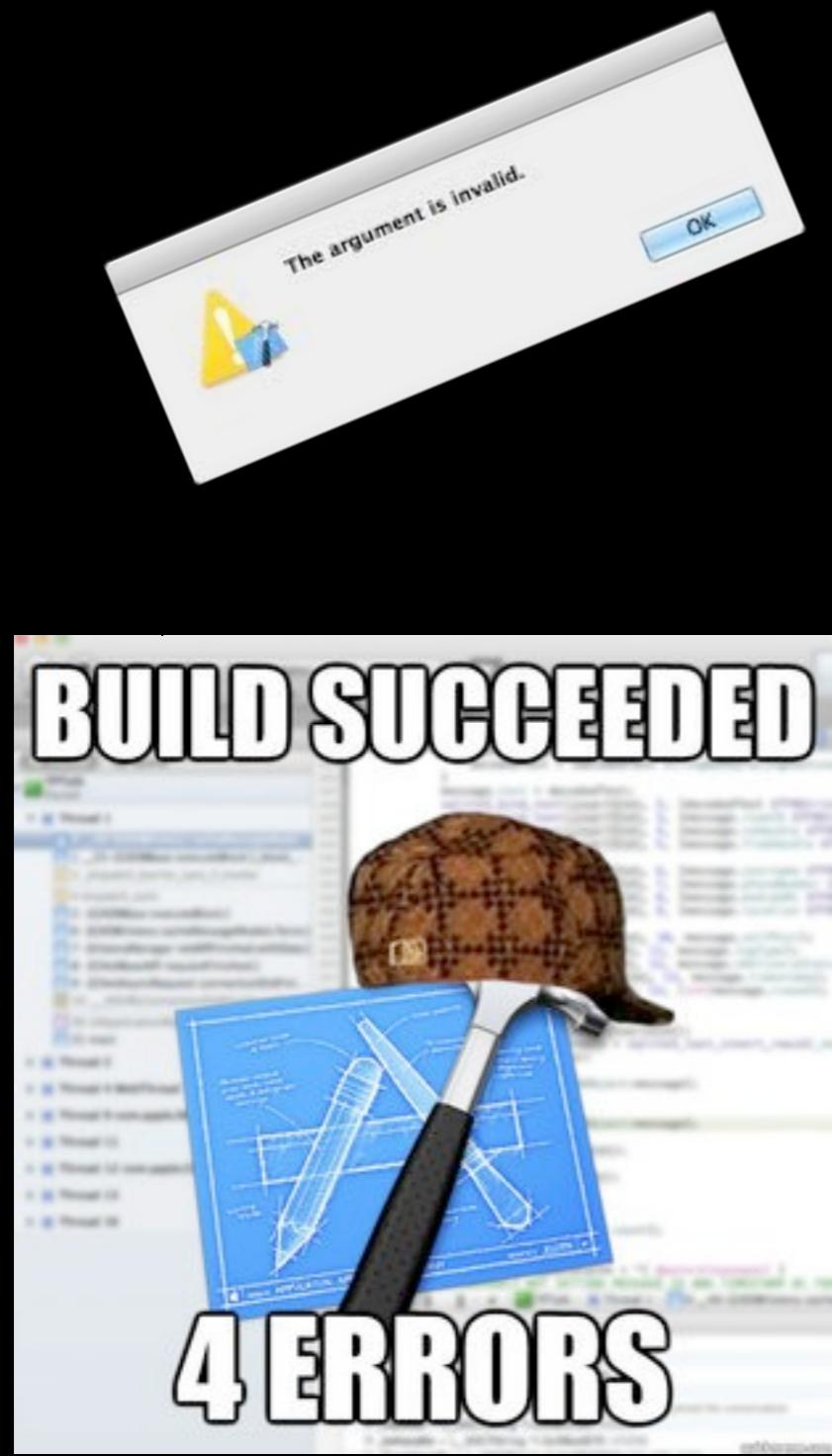
Xcode





I got it you just
press here and
there...

Xcode is not . . .







FREEDOM

Shift
↑

Ctrl

Editors



[Mac] Vim

A screenshot of the Vim text editor on a Mac. The title bar reads "app_delegate.rb + (~/Desktop/demo/app) - VIM1". The main window displays the following Ruby code:

```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3     @window = UIWindow.alloc.initWithFrame UIScreen mainScreen.bounds
4
5     evc = EpisodesViewController.alloc.init
6     nav = UINavigationController.alloc.initWithRootViewController evc
7     @window.rootViewController = nav
8     @window.makeTextWritingDirectionRightToLeft: []
9     true makeKeyAndVisible
10    makeKeyWindow
11    makeObjectsPerformSelector:
12    makeObjectsPerformSelector:withObject:
13    makeTextWritingDirectionLeftToRight:
14    makeTextWritingDirectionRightToLeft:
```

The line "makeTextWritingDirectionRightToLeft: []" is highlighted with a yellow rectangle, indicating a keyword completion suggestion. The status bar at the bottom shows "INSERT" mode, the file path "master > app/app_delegate.rb ?? +", the encoding "unix < utf-8 < ruby", the file size "73%", and the current time "8:17". A message "Keyword completion (^N^P) match 1 of 6" is also visible.

Sublime Text 2

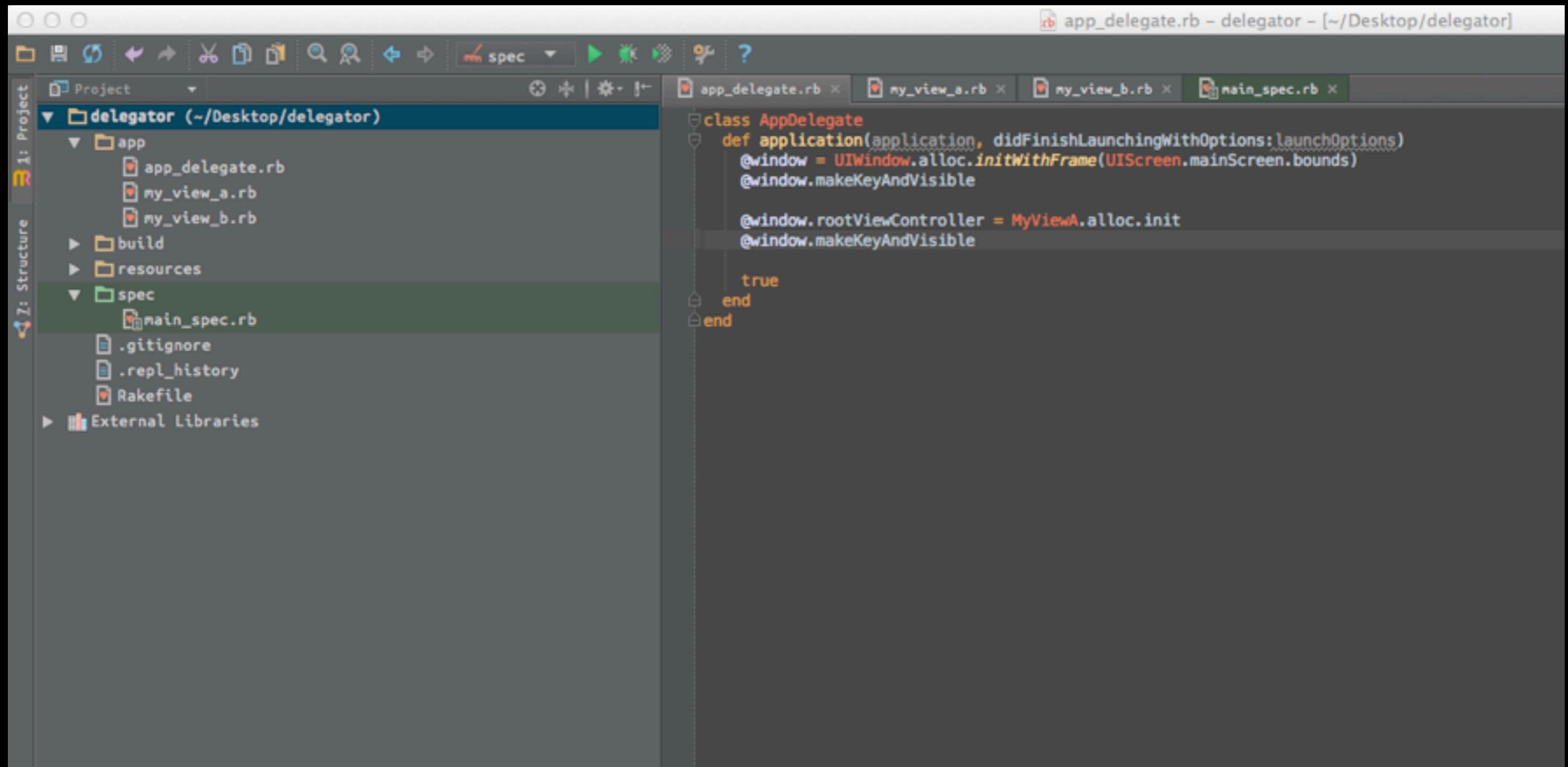
The screenshot shows the Sublime Text 2 interface with the following details:

- Open Files:** app_delegate.rb
- Folders:** demo (which contains app, api.rb, app_delegate.rb, episode.rb, episodes_view_controller.rb, build, resources, spec, .gitignore, .repl_history, Gemfile, Gemfile.lock, Rakefile, and tags)
- Active File:** episodes_view_controller.rb — demo
- Code Content:** The code is a Ruby class definition for `EpisodesViewController`. It includes methods for `init`, `viewDidLoad`, `tableView:numberOfRowsInSection:`, and `tableView:cellForRowAtIndexPath:`. It uses `ApiClient.fetch_episodes` to fetch data and `UITableViewController` to handle table view logic.

```
1 class EpisodesViewController < UITableViewController
2   def init
3     super initWithStyle(UITableViewStylePlain)
4   end
5
6   def viewDidLoad
7     super
8     self.title = "Episodes"
9     @episodes ||= []
10
11   ApiClient.fetch_episodes do |sucess, episodes|
12     if sucess
13       @episodes = episodes
14       p "Received #{@episodes.length} episodes"
15       self.tableView.reloadData
16     else
17       App.alert("Ooops!")
18     end
19   end
20 end
21
22 def tableView(tableView, numberOfRowsInSection:section)
23   @episodes.count
24 end
25
26 def tableView(tableView, cellForRowAtIndexPath:indexPath)
27   cell_id = "cell"
28   cell = tableView.dequeueReusableCellWithIdentifier cell_id
29   if cell.nil?
30     cell = UITableViewCell.alloc.initWithStyle UITableViewCellStyleDefault, reuseIdentifier:cell_id
31   end
32   episodes = @episodes[indexPath.row]
33   cell.textLabel.text = episodes.title
34   cell
35 end
36
37 end
38
```

- Status Bar:** Line 1, Column 1, Spaces: 2, RubyMotion

RubyMine



Or just... .



```
$ motion
```

Usage:

```
motion [-h, --help]  
motion [-v, --version]  
motion <command> [<args...>]
```

Commands:

account	Access the software license account
activate	Activate the software license
create	Create a new project
ri	Display API reference
support	Create a support ticket
update	Update the software
joybox:generate	Joybox: Class Generator
joybox:retina	Joybox: Retina Generator

```
$ motion create demoapp
```

```
demoapp
└── Rakefile
    └── app
        └── app_delegate.rb
    └── resources
        └── Default-568h@2x.png
    └── spec
        └── main_spec.rb
```



3 directories, 4 files

```
demoapp
└── demoapp
    ├── AppDelegate.h
    ├── AppDelegate.m
    ├── Default-568h@2x.png
    ├── Default.png
    ├── Default@2x.png
    ├── demoapp-Info.plist
    ├── demoapp-Prefix.pch
    └── en.lproj
        └── InfoPlist.strings
    └── main.m
└── demoapp.xcodeproj
    ├── project.pbxproj
    ├── project.xcworkspace
    │   └── contents.xcworkspacedata
    └── xcuserdata
        └── yaakovgamlieL.xcuserdata
            └── UserInterfaceState.xcuserstate
└── xcuserdata
    └── yaakovgamlieL.xcuserdata
        └── xcscmchemes
            └── demoapp.xcscheme
            └── xcscmhemanagement.plist
```



9 directories, 14 files

Let's begin with delegates!

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    // Override point for customization after application launch.
    self.window.backgroundColor = [UIColor whiteColor];
    [self.window makeKeyAndVisible];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types
    // of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application
    // and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough
    // application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of
    applicationWillTerminate: when the user quits.
}


```

etc . . .

@end



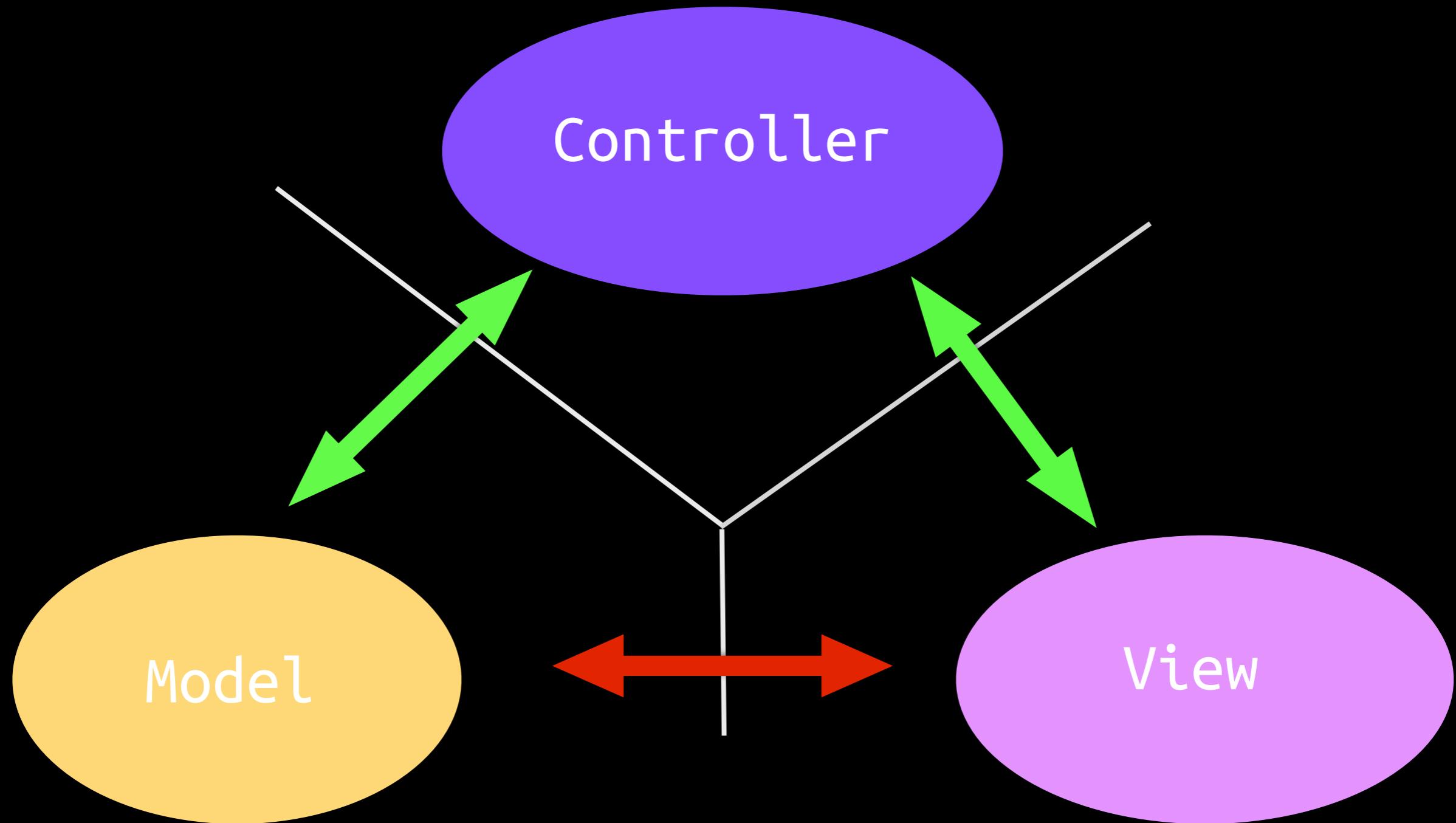
Delegates

Maybe callbacks?

Well, not really

Demo

MVC



Gems

bundler.io

Bundler: The best way to manage a Ruby application's gems

Docs: v0.9 v1.0 v1.1 v1.2 v1.3

Fork me on GitHub



Bundler

The best way to manage your application's dependencies

What is Bundler?

Bundler maintains a consistent environment for ruby applications. It tracks an application's code and the rubygems it needs to run, so that an application will always have the exact gems (and versions) that it needs to run.

Would you like to

[Get started](#) [Report a bug](#) [See what's new](#) [Read documentation](#)

Getting Started

Getting started with bundler is easy! Open a terminal window and run this command:

```
$ gem install bundler
```

cocoapods.org

CocoaPods: The Objective-C Library Manager

cocoapods.org Reader

CocoaPods Either OS iOS & OSX iOS OSX pod name, version, author, deps...

The best way to manage library dependencies in Objective-C projects.

Install

CocoaPods is distributed as a **ruby gem**, installing it is as easy as running the following commands in the terminal:

```
$ [sudo] gem install cocoapods  
$ pod setup
```

Now that you've got CocoaPods installed it's time to **get started**.

Get started

Search for pods by name and description.

Then list the dependencies in a text file named `Podfile` in your Xcode project directory:

```
$ edit Podfile  
platform :ios, '6.0'  
pod 'JSONKit',      '~> 1.4'  
pod 'Reachability', '~> 3.0.0'
```

Now you can install the dependencies in your project:

```
$ pod install
```

Make sure to always open the Xcode workspace instead of the project file when building your project:

```
$ open App.xcworkspace
```

Now you can import your dependencies e.g.:

```
#import <Reachability.h>
```

bubblewrap.io

BubbleWrap – A Cocoa wrapper by Matt Aimonetti

Reader

BubbleWrap

Cocoa wrappers and helpers for RubyMotion.

[View the Project on GitHub](#)
`rubymotion/BubbleWrap`

[Read the RDoc](#)
`rdoc.info`

[Get the Ruby Gem](#)
`gem 'bubble-wrap'`

[Join the discussion](#)
`bubblewrap@googlegroups.com`

[Using BubbleWrap](#)
A guide to getting started with BubbleWrap

[Hacking BubbleWrap](#)
A guide to working on BubbleWrap

[Gem Howto](#)
Building RubyMotion gems with BW

Maintained by:

 Clay Allsopp
`@clayallsopp`

 Dylan Markow
`@dmarkow`

 James Harton
`@jamesotron`

 Jan Weinkauf
`@janweinkauf`

 Marin Usaj
`@mneorr`

 Francis Chong
`@siuying`

BubbleWrap for RubyMotion

A collection of (tested) helpers and wrappers used to wrap CocoaTouch code and provide more Ruby like APIs.

[BubbleWrap website](#) [BubbleWrap mailing list](#)

Installation

```
gem install bubble-wrap
```

Setup

1. Edit the **Rakefile** of your RubyMotion project and add the following require line:

```
require 'bubble-wrap'
```

BubbleWrap is split into multiple modules so that you can easily choose which parts are included at compile-time.

The above example requires the **core** and **http** modules. If you wish to only include the core modules use the following line of code instead:

```
require 'bubble-wrap/core'
```

If you wish to only include the **HTTP** wrapper:

```
require 'bubble-wrap/http'
```

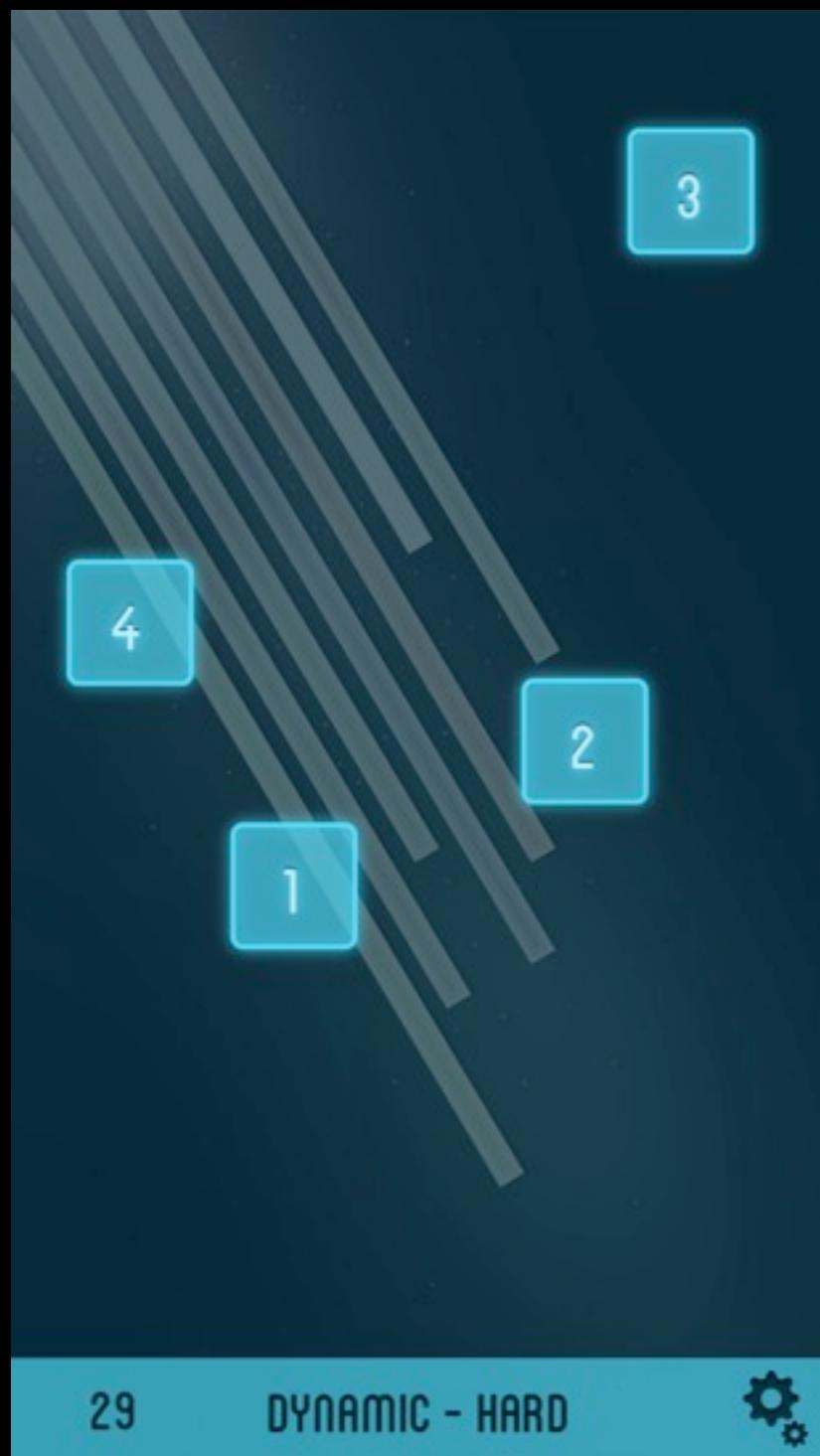
Joybox.io

The screenshot shows a web browser window with the title bar "Untitled — Edited". The address bar displays "Joybox - Videogames are the passion, Ruby the language and RubyMotion the platform" and the URL "joybox.io". The browser interface includes standard controls like back, forward, and search.

The website itself has a dark red header with the "Joybox" logo on the left and navigation links: GET STARTED, DOCUMENTATION, LEARN THIS, HIGH SCORES, VERSION HISTORY, and HELP. The main content area features large white text: "Video games are the **passion**, **Ruby** the language and **RubyMotion** the platform". Below this is a stylized illustration of a red robot or character in a futuristic setting with tall poles and a starry background. A yellow button at the bottom center says "START HERE".

APPS

<http://www.rubymotion.com/apps/>



ULTRA REACTION

RESPONSIVE
TEST YOUR REACTION SPEED

DYNAMIC
IMPROVE YOUR SPEED & FOCUS

ENDLESS
REACTION SPEED + ENDURANCE

ACHIEVEMENTS

TARGETS HIT

0

1 2 3 4 5 6 7 8 9 10

I HOPE NO ONE WERE WATCHING THAT. AWFUL

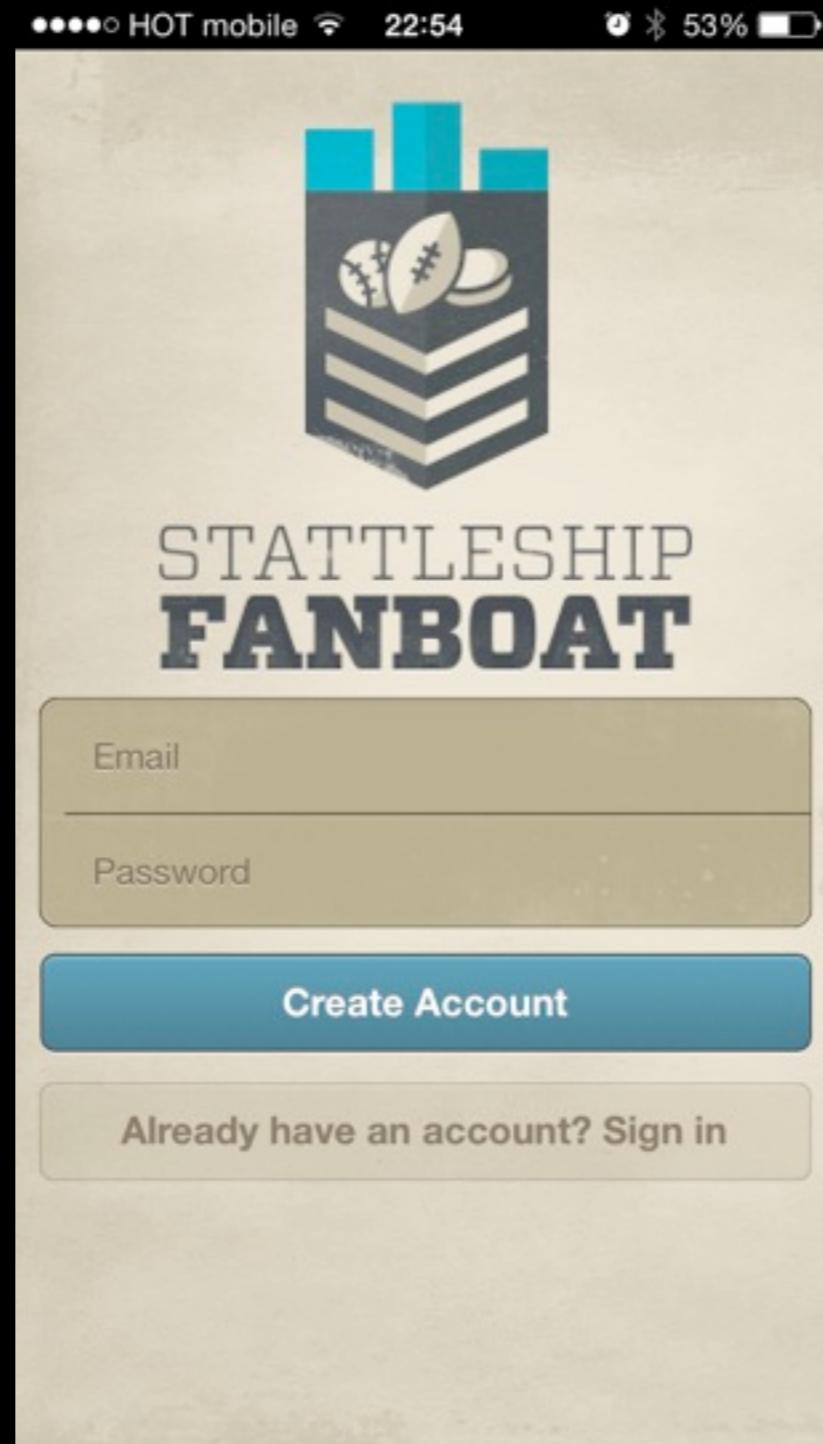
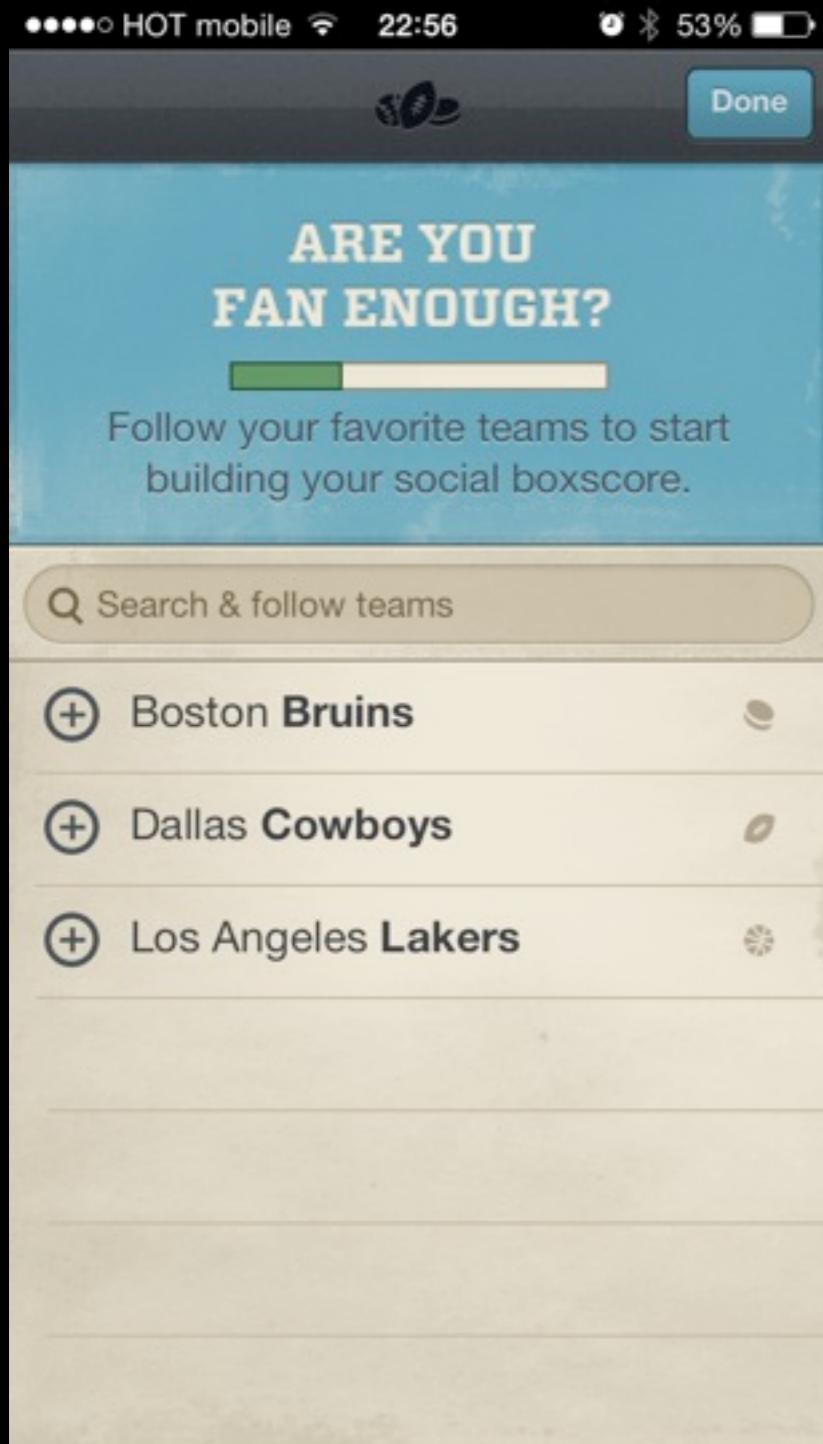
NEW RECORD: 0
RATED 1/10

TWEET

LOADING...

LEADERBOARD

BACK TRY AGAIN



••••• HOT mobile 22:57 53%

Fanboat Game

July 11	R	H	E
NY YANKEES	8	9	0
KC ROYALS	1	6	0

★ Favorite Share

Brett Gardner was hit by 2 hit by pitches NYY

Lyle Overbay hit 1 grand slam NYY

Robinson Cano had 3 hits NYY

Robinson Cano hit 3 RBI NYY

Lyle Overbay hit 4 RBI NYY

Ivan Nova recorded 24 outs NYY

Wade Davis allowed 8 runs KAN

Wade Davis allowed 8 runs

This screenshot shows the game summary screen for the Fanboat app. It displays a table of recent games, with the top game being the New York Yankees vs. Kansas City Royals on July 11, 2013. The table includes columns for the date, team names, and statistics (R, H, E). Below the table, there are links for "Favorite" and "Share". The main content area lists key events from the game, such as Brett Gardner hitting by pitch twice, Lyle Overbay hitting a grand slam, and Robinson Cano having three hits and three RBIs. Other events listed include Ivan Nova's 24 outs and Wade Davis allowing eight runs.

More gems..

UI

<https://github.com/rubymotion/teacup>

DSL

<https://github.com/clearsightstudio/ProMotion>

Networking

<https://github.com/clayallsopp/afmotion>

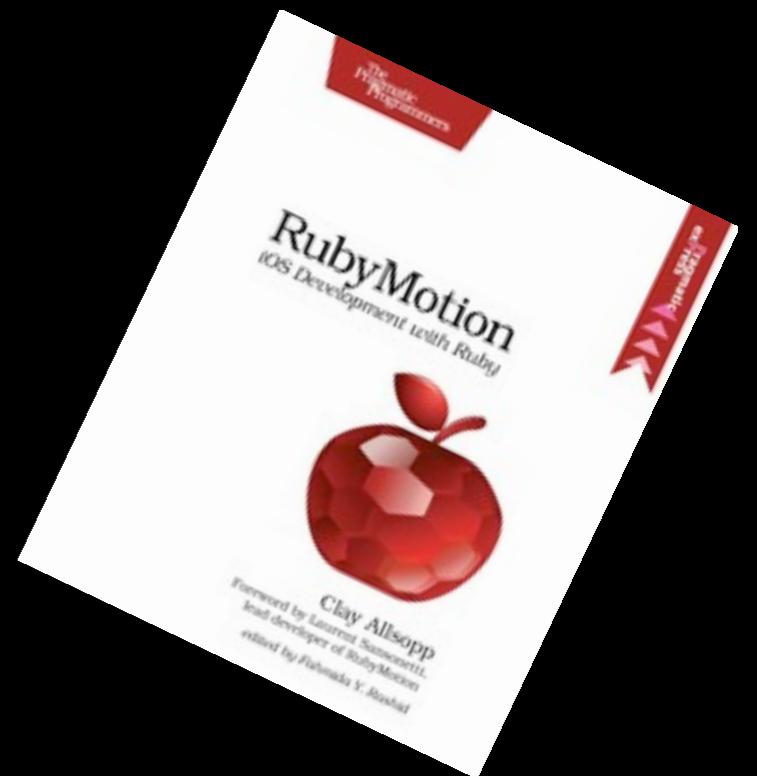
Good resources

Blogs

<http://iconoclastlabs.com> #this guys are going RM in production

Books

<http://rubymotion-tutorial.com>



Videos

<https://vimeo.com/hipbyte>

We get updates very fast!

 **RubyMotion** @RubyMotion 1d

RubyMotion 2.6 is out, with support for iOS 7.0 beta 4 and a few fixes in the retain cycle detector. Enjoy!

= RubyMotion 2.4 =

- * Added support for iOS 7.0 Beta 3 and Xcode 5.0 DP 3.
- * Fixed a long-standing set of memory-related bugs related to lambdas. Dynamic variables (shared by lambdas and calling scope) are now allocated as heap memory inside the Proc data structure. Local variables are properly synchronized. Proc objects are properly reclaimed by the system once they are no longer used. Changes have been added to both the compiler and the runtime.
- * [iOS] Fixed a bug in 'rake device' where it could not Xcode 5.x with a deployment target less than iOS 7.0.
- * [iOS] Reduced the size of the Default-568h@2x.png file Laborde for the patch (pull request #100).
- * [iOS] Fixed a bug where 'rake retina=3.5' would not work.
- * [iOS] Fixed a bug where 'rake simulator' would re-launch with Xcode 5.x every time.
- * [OSX] Fixed a bug in the build system where APIs of 32-bit would not be properly handled in 64-bit mode.
- * Fixed a memory crasher when trying to use the old value of an instance variable slot as a local variable.
- * Fixed memory leaks in NSMutableArray#insert.
- * Fixed memory leaks when parsing bridgesupport file during compilation.
- * Fixed a bug in NSMutableDictionary#delete where it will cause a crash if it would use the result of NSMutableDictionary#delete on the receiver object which was created by NSJSONReadingMutableContainers.
- * Fixed a bug where 3rd-party libraries would fail to build if the file path of the app contained space characters.
- * Fixed a bug in Kernel#Array() where it would return an incorrected value if an NSArray object was passed as argument.
- * Fixed a compiler bug where 'not precompiled' errors would sometimes happen at runtime when using APIs dealing with BOOL types.
- * Added a block to setup controller in the spec API. Thanks to Katsuyoshi Ito for the patch (pull request #104).
- * Fixed the 'rake build' task to honor the 'keep_temps' environment variable. If set, temporary files (such as LLVM bitcode and assembly) will not be removed from the build directory. To be used for debugging only.

(See the '/Library/RubyMotion/NEWS' file for all changes.)

yaakovs-MacBook-Pro:~ yaakovgamlieis\$ |

 **RubyMotion** @RubyMotion

RubyMotion 2.4 is released, bringing support for iOS 7.0 beta 3 and also fixes for long-standing memory-related crashes (ex. RM-3). Enjoy!

Retweeted by Andrew Havens

released twice, leading to a crash.
d a bug in the build system where app.deployment_target and
sdk_version were compared as text. Thanks to Andrew Babichev for the
h (pull request #106).
ged the 'rake simulator' task to print a message when an application
es due to a signal (ex. SIGSEGV) and suggest the user to read the
h report file that has been generated by the system.
d the 'rake crashlog' task to automatically open the latest crash
rt file that was generated by the system for the current app.
d a bug in the REPL where using the <tab> key to enable auto-completion
d result in a crash due to a missing symbol.
d a bug where an object implementing the NSCoder protocol as well as
t and #initialize(arg) methods would cause a crash during creation.
) Fixed a warning message from the linker that would be displayed when
g Xcode 5 to build the app.
) Fixed a bug in the debugger where built-in debug commands weren't
lled within the simulator.
) Fixed a bug in the build system where an app compiled for iOS 5.x
d immediately crash on the device.
) Fixed a bug where the NSUIntegerMax constant was not properly
sed when targeting iOS 7.x.

Community

Google groups

<https://groups.google.com/forum/#!forum/rubymotion>

IRC #rubymotion at freenode

Twitter @RubyMotion

Not every Ruby code needs to
look like Rails



Now go out there and
build something
awesome!

Thank you!

?