RubyMotion:
The sleeper has awakened!

Nick Quaranto - @qrush
REMOTE
<3 RubyMotion!
Basecamp for iPhone
awakening
I kept getting stuck.
RESURFACE ICE, FRANK ZAMBONI STYLE!

In 1949 Frank Zamboni built the first working ice resurfacing machine. It revolutionized the ice rink business by allowing for continuous ice use during a season.
RESURFACE ICE,
FRANK ZAMBONI STYLE!

In 1949 Frank Zamboni built the first working resurfacer for ice rinks.
Break the cycle!
Files - the first app
Log in to your Basecamp

Don’t have Basecamp? Try it free today.
JM #3: turn message into to-do

Posted by Javan Makhmali

JM #3: turn message into to-do

Discussion 1 comment

Javan Makhmali – My implementation latches on to move operations to turn a message into a to-do and put it in a list in the same project. The subject of the message becomes the to-do content, and the body becomes a comment on the to-do. (This may be another case for a description field on to-dos.)
and got rejected

Rejected.
Here's what's been happening in your projects.

**Today**

Mar 11

---

**Improve Conversion Funnel**

---

**Jamie D. commented on** Public Radio

Good Evening Jamie- I hope your weekend went well. Attached is a proposal for a four-week sponsorship of the This American Life podcast a...

---

**Add a comment...**

---

**Nick Quaranto said...**

8 hours ago

Deposit submitted! I also gave them some CoworkBuffalo stickers. Who doesn't like stickers?

---

**Mark Josef said...**

4 hours ago

Stickers are good... PINS ARE BETTER: [http://www.qualitylapelpins.com/custom_lapel_pins.php](http://www.qualitylapelpins.com/custom_lapel_pins.php)
I could not have done this without RubyMotion
was a roadblock
what awakened?
style
I write to learn about software, and I learn software by writing it.
def foo_bar(a, b)
def foo_bar(a, b: c)
classic dialect
ruby dialect
eventmachine dialect
objective-c dialect
community
gem 'bubble-wrap'
gem 'bubble-wrap'
HTTP requests
NSNotificationCenter
Quick persistence
Lots of shortcuts
gem 'bubble-wrap'
gem 'motion-settings-bundle'
# in your Rakefile
require 'motion-settings-bundle'

Motion::SettingsBundle.setup do |app|
  app.text "Version", key: "version", default: "1.0.0"
  app.toggle "Awesome?", key: "awesomeOn", default: true

  app.child "Acknowledgements" do |ack|
    ack.child "AwesomeOSSLibrary" do |lic|
      lic.group "Copyright 2013 AwesomeOSSContributor"
      lic.group "A bunch of legal stuff"
    end
  end
end
Create Constraints Programmatically Using an ASCII-Art Format String

You represent constraints using instances of `NSLayoutConstraint`. To create constraints, you typically use `constraintsWithVisualFormat:options:metrics:views:`. The first argument is an ASCII-art–inspired visual format string that provides a visual representation of the layout you want to describe. A view is represented in square brackets, and a connection between views is represented using a hyphen (or two hyphens separated by a number to represent the number of points apart the views should be). Examples and a grammar for the visual format language are given in “Visual Format Language” (page 13).

For example, for this layout
Switch Basecamp accounts

Need help with Basecamp?
gem 'motion-layout'

DSL for NSLayoutConstraint

iOS6+ Only

Extracted from Basecamp.app

No tests, but examples!
Motion::Layout.new do |lay|
  lay.view self.view.tableFooterView
  lay.subviews "switch" => @switch, "help" => @help
  lay.vertical "|-15-[switch]-10-[help(==switch)]-15-|
  lay.horizontal "|-10-[switch]-10-|
  lay.horizontal "|-10-[help]-10-|
end
resource
Source files

The simplest way to add the AwesomeObjCThing to your project is by adding the source files to your project.

AwesomeObjCThing.m

1. Download the latest code version or add the repository to your project.
2. Open your project in Xcode, then drag and drop "AwesomeObjCThing" into your project. Make sure to select Combine in the Add to Project dialog.
3. Include AwesomeObjCThing wherever you need it in your code.
Open your project in Xcode,
NO.
yours?
thanks