USE RUBY TO WRITE (AND TEST) YOUR NEXT ANDROID APP

Joel Byler
Software Craftsman
@joelbyler

June 1, 2013
Pittsburgh TechFest
La Roche College

WHO AM I?

JOEL BYLER

- Software Craftsman
- Organizer for CleRb
- Enterprise Java Developer
- Relatively new to Ruby
- ... and Android


Dude...
Wait, what?

NEW TO ANDROID AND RUBY???

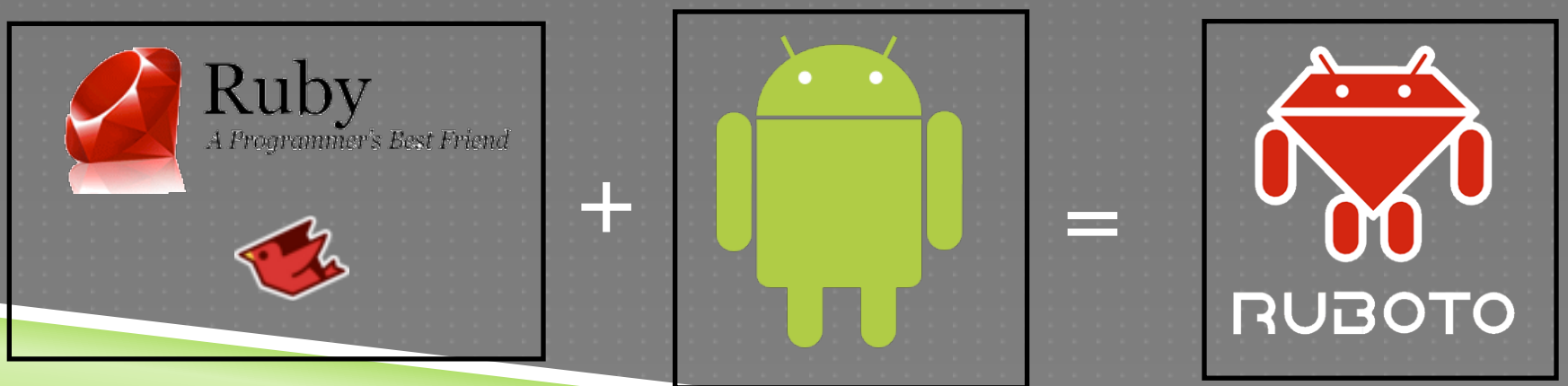
- Software Craftsmanship
  - Reduce risk with TDD / ATDD
- <3 Ruby and want to learn more
- <3 Mobile apps, who doesn't?

Warning: this may turn out to be a lot of code to absorb in a short amount of time. Slides and code will be made publicly available

RUBY ON ANDROID USING RUBOTO

- JRuby optimized for the Android OS
  - JRuby is a JVM language
- Android uses something like the JVM
  - but actually Dalvik VM

<<< MAGIC >>>



IRB ON YOUR ANDROID DEVICE

- Interactive Ruby Shell
- Edit and run scripts in Android

- demo...

PERFORMANCE CONSIDERATIONS

Other than initial startup to load JRuby, the performance of the app appears to be as good as many native Android apps

RUBOTO CORE

- A separate app available on the Google Play Store
- Allows Ruboto Runtime (JRuby on Dalvik VM) to be shared
- An alternative would be to use the --with-jruby option

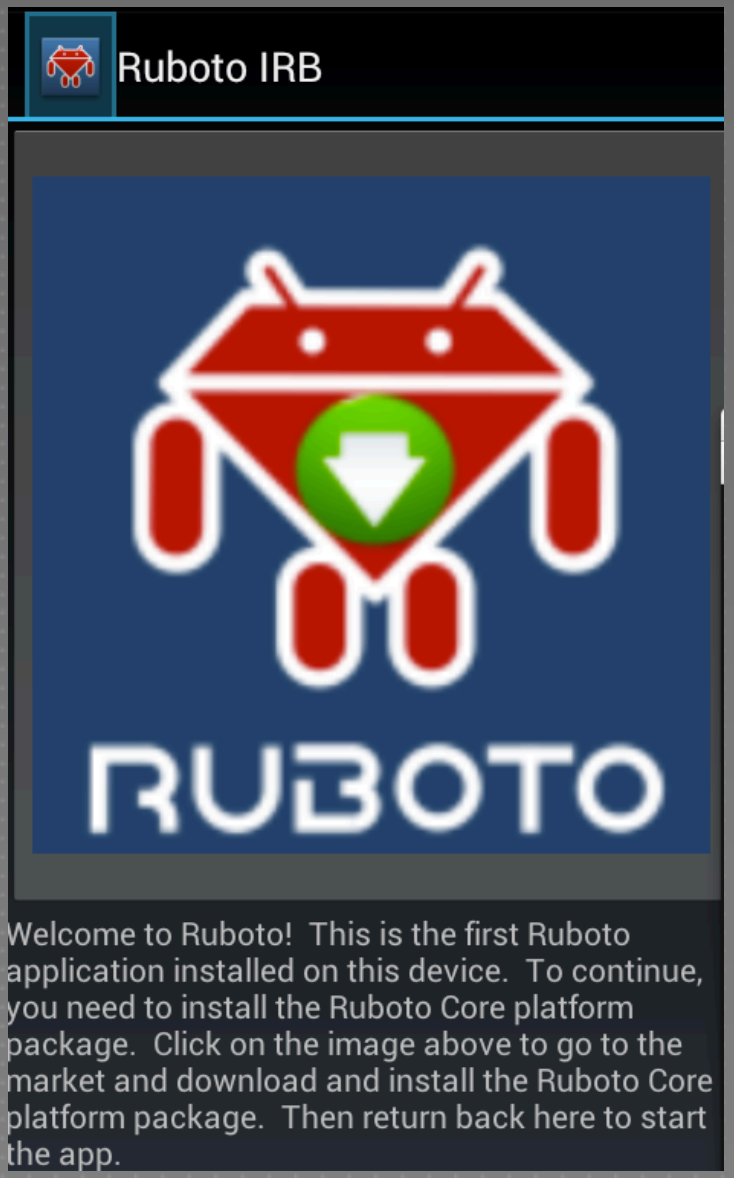
Decisions, decisions...

HOW DOES THIS MAKE YOU FEEL?

This is what your users will see if they don't already have the Ruboto Core installed

or

Using the --with-jruby option will add ~10mb to the size of your app



CREATE A NEW PROJECT

Ruboto Application Generator

```
$ ruboto gen app --package com.leandog.mastermind.ruboto
    --name MasterMindRuboto --target android-15
    --with-jruby --path=RubotoMasterMind
```

This will generate application with
'What hath Matz wrought?' sample code.

RUBOTO CLASS GENERATOR

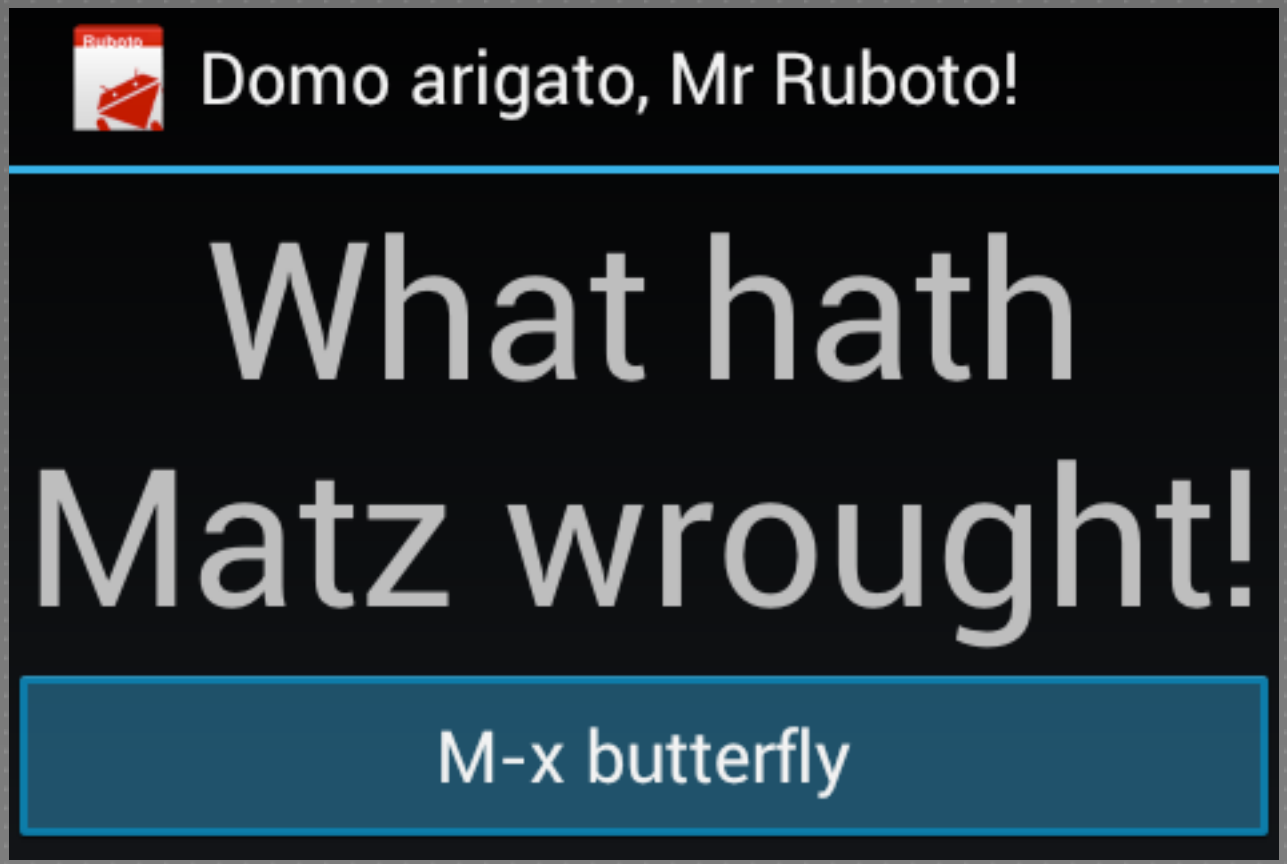
Ruboto Class Generator

```
$ ruboto gen class Activity --name MasterMindMainActivity
```

This will generate code for the activity and add it to the project manifest.

You can also use this to generate a BroadcastReceiver or Service class

YOUR GENERATED APP



YOUR GENERATED CODE (RUBY .RB)

```ruby
class MasterMindRubotoActivity
  def onCreate(bundle)
    super
    set_title 'Domo arigato, Mr Ruboto!'
    self.content_view =
        linear_layout :orientation => :vertical do
          @text_view = text_view :text => 'What hath Matz wrought?', :id => 42,
          button :text => 'M-x butterfly', :width => :match_parent, :id => 43, :
        end
  rescue
    puts "Exception creating activity: #{$!}"
    puts $!.backtrace.join("\n")
  end

  private

  def butterfly
    @text_view.text = 'What hath Matz wrought!'
    toast 'Flipped a bit via butterfly'
  end
```

YOUR GENERATED TEST (ALSO .RB)

```
1  activity Java::com.leandog.demo.ruboto.MasterMindRubotoActivity
2
3  setup do |activity|
4    start = Time.now
5    loop do
6      @text_view = activity.findViewById(42)
7      break if @text_view || (Time.now - start > 60)
8      sleep 1
9    end
10   assert @text_view
11 end
12
13 test('initial setup') do |activity|
14   assert_equal 'What hath Matz wrought?', @text_view.text
15 end
16
17 test('button changes text') do |activity|
18   button = activity.findViewById(43)
19   button.performClick
20   assert_equal 'What hath Matz wrought!', @text_view.text
21 end
```

RAKE INSTEAD OF ANT

```
$ rake install start
```

Install the project on a device or emulator and start it running.

```
$ ant debug
$ adb install example.apk
$ adb shell am start -n com.foo.bar...
```

```
$ rake update_scripts
```

Install the project on a device or emulator and start it running.

```
$ adb install example.apk
```

`$ rake -T` List other available rake tasks

ADD IN THE ATDD

```
$ gem install testgen
```

This is one of @chzy 's testing gems

```
$ testgen project yourapp --with-gametel
```

This will add everything you need to use cucumber to test your Ruboto app

NOW FOR SOME UNIT TESTS

`$ mkdir spec` This is where your specs will live

Add rspec to Rakefile (created by testgen)

```
1  require 'cucumber/rake/task'
2  require 'rspec/core/rake_task'
3
4  Cucumber::Rake::Task.new(:features) do |t|
5    t.profile = 'default'
6  end
7
8  RSpec::Core::RakeTask.new(:spec) do |spec|
9    spec.ruby_opts = "-I lib:spec"
10   spec.pattern = 'spec/**/*_spec.rb'
11 end
12
13 task :default => [:test, :spec, :features]
```

UNIT TESTING

Use rspec to test your ruby code, just like you normally would.

```ruby
21    context "initialized game" do
22      let(:mastermind) { MasterMind.new 4, 2, 1, 9 }
23
24      it "can give feedback on a guess" do
25        result = mastermind.guess(1, 2, 3, 4)
26        result.should_not be_nil
27      end
28
```

rspec

```java
5   public class MasterMindTest extends TestCase{
6     private MasterMind masterMind;
7     @Override
8     protected void setUp() throws Exception {
9       masterMind = new MasterMind();
10    }
11    public void testThatMasterMindCanGiveFeedbackOnAGuess() {
12      Guess result = masterMind.guess(1, 2, 3, 4);
13      assertNotNull(result);
```

JUnit 3

rake spec

INTEGRATION TEST?

Instruments and installs app on device when running

```
3   setup do |activity|
4     start = Time.now
5     loop do
6       @text_view = activity.findViewById(MasterMindMainActivity::TEXT_MESSAGE_VIEWID)
7       break if @text_view || (Time.now - start > 60)
8       sleep 1
9     end
10    assert @text_view
11  end
12
13  test('initial setup') do |activity|
14    assert_equal 'Enter four numbers below and submit your guess', @text_view.text
15  end
16
17  test('button changes text') do |activity|
18    button = activity.findViewById(MasterMindMainActivity::BUTTON_SUBMIT_GUESS)
19    button.performClick
20    assert @text_view.text.match 'You have (.*) numbers and (.*) positions correct.'
21  end
```

similar to ActivityInstrumentationTestCase2

rake test

ACCEPTANCE TESTING

Cucumber
Gametel
Brazenhead
ADB gem

Feature: Input Screen

Scenario: Welcome the user
When the application launches
Then I see "Enter four numbers below and submit your guess"

Scenario: User can guess
Given I enter 4 numbers
When I press the submit button
Then I see "Nice guess!"

```
Then(/^I enter 4 numbers$/) do
  on(MainScreen).guess 1, 2, 3, 4
end
```

```
class MainScreen
 include Gametel

 text(:number1, :index => 0)
 text(:number2, :index => 1)
 text(:number3, :index => 2)
 text(:number4, :index => 3)

 def guess (first, second, third, forth)
  number1 = first.to_s
  number2 = second.to_s
  number3 = third.to_s
  number4 = forth.to_s
 end
end
```

rake features

YOUR RUBOTO CLASS FILES

- They're just .rb files
- Allows you to make changes without a full rebuild
- fast feedback
- Ruboto uses ruby classes that are backed by java classes for improved performance.
- Can still reference Java resources

example:

```
R::string::header_text
```

QUICK TOUR OF THE CODE

* Time permitting


HALP

THE RUBOTO COMMUNITY IS GROWING

Some of these projects are still young or have a small user base.

- They will probably continue to change (improve)
- Ruboto is only version 0.12

Example:

Wasn't able to add a menu to the main activity. Instead I had to wrap it with the 'launch' activity.

The good news is that they could definitely use YOUR HELP!

Anyone up for trying to make the debugger work?

CONCLUSION

Ruboto is a great Open Source project with a growing community that promises to get even better.

Yes, it does increase your app's startup time slightly, but offers a great dynamic nature for rapidly changing apps.

Virtually all code can be Ruby, but can also reference Java resources if you so desire.

RESOURCES

Sample Code
https://github.com/joelbyler/Ruboto-MasterMind

Ruboto
http://ruboto.org/

ATDD Training Resource
https://leanpub.com/cucumber_and_cheese

More In-Depth Personal Training
http://www.TestAutomationBootCamp.com/

DOMO ARIGATO MR. ROBOTO!

Fun fact:
domo arigato actually means
"Thanks a lot!"
in Japanese
http://en.wikipedia.org/wiki/Domo_arigato

So....

Domo Arigato!

THAT'S A WRAP!

Joel Byler
@joelbyler

LeanDog

COME FLOAT WITH US!

Located on a 10,000 square foot boat on the Cleveland waterfront, our team exemplifies how Agile values & principles are practiced. Come float, deliver, and learn with us, or leverage our expertise to help you change your company culture.

@leandog

www.leandog.com

LeanDog