

Demo Project

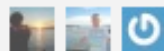
build status passing

Current Sprint

Sprint Planning

IN PROGRESS

1.o Add User Story Box in Sprint Planning Mode



Show Description



Complete

1.p JavaScript Pop-Up for more Details



Show Description

Complete

4.f Accuracy of Estimations



Show Description



Complete

SPRINT BACKLOG

Prepare a talk about Shoes

Show Description



Start

We know that we can use Ruby for Web Applications...

1.o Add User Story Box in Sprint Pla...

4.f Accuracy of Estimations

SUSPENDED

2 (Priority)

2 (Estimation)

Description

Add user story box in sprint planning mode is used to create a new user story on the sprint planning page.

Participants



Show | Edit



Complete

Partner

```
( Sometimes I wonder if I'm in my right )
( mind. Then it passes off and I'm as   )
( intelligent as ever.                  )
(                                       )
( -- Samuel Beckett, "Endgame"         )
```

```
o
o
```

```
{~.~.~}
( Y )
()~*~()
( )-( )
```

```
tobi@speedy ~ $ cd github/shoes4/
```

```
Using /home/tobi/.rvm/gems/jruby-1.6.7 with gemset shoes
```

```
tobi@speedy ~/github/shoes4 $ rake spec
```

```
jruby --debug --1.9 -Ispec -S ./bin/rspec --tty spec/shoes/flow_spec.rb spec/shoes/sound_spec.rb spec/shoes/animation_spec.rb spec/shoes/oval_spec.rb spec/shoes/color_spec.rb spec/shoes/radio_spec.rb spec/shoes/button_spec.rb spec/shoes/shape_spec.rb spec/shoes/list_box_spec.rb spec/shoes/framework_learning_spec.rb spec/shoes/logger_spec.rb spec/shoes/line_spec.rb spec/shoes/element_methods_spec.rb spec/shoes/app_spec.rb spec/shoes/configuration_spec.rb spec/shoes/text_block_spec.rb spec/shoes/logger/log4j_spec.rb spec/shoes/logger/ruby_spec.rb
```

```
Shoes::Logger
```

```
.get
  retrieves a registered logger
.register
  allows new loggers to be registered
```

```
Shoes::Line
```

```
specified right-to-left, top-to-bottom
behaves like basic line
```

```
should be a kind of Shoes::Line
```

```
width
  should eq 90
```

```
height
  should eq 45
```

```
left
  should eq 10
```

```
top
  should eq 15
```

```
specified right-to-left, bottom-to-top
behaves like basic line
```

```
should be a kind of Shoes::Line
```

```
width
  should eq 90
```

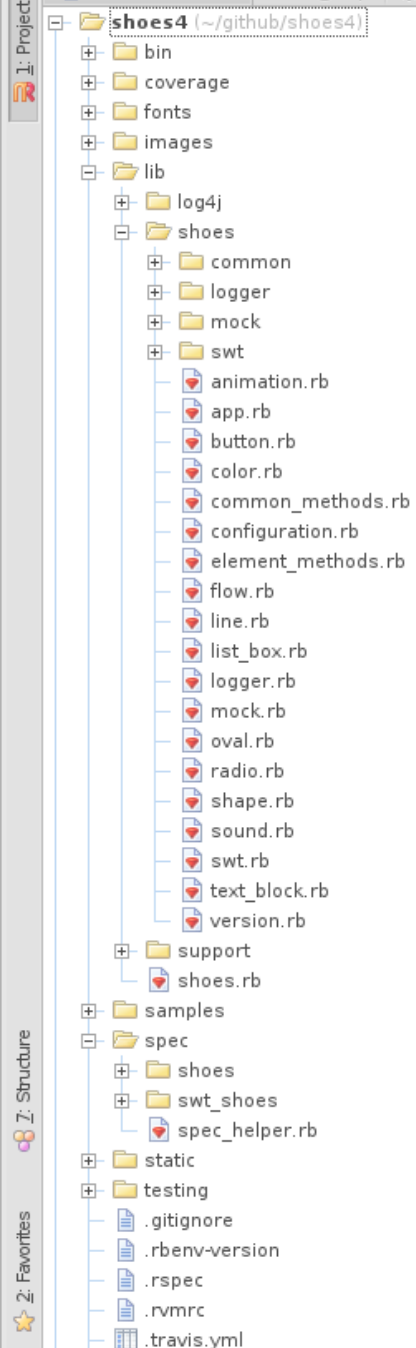
```
height
  should eq 45
```

```
left
  should eq 10
```

```
top
  should eq 15
```

```
created left-to-right, top-to-bottom
```

Or Console Applications...



```
1 require 'shoes/animation'
2 require 'shoes/sound'
3 require 'shoes/button'
4 require 'shoes/color'
5 require 'shoes/flow'
6 require 'shoes/line'
7 require 'shoes/oval'
8 require 'shoes/shape'
9 require 'shoes/text_block'
10 require 'shoes/list_box'
11 require 'shoes/radio'
12
13 module Shoes
14   # Methods for creating and manipulating Shoes elements
15   #
16   # Requirements
17   #
18   # Including classes must provide:
19   #
20   # @style - a hash of styles
21   module ElementMethods
22     #def stack(opts={}, &blk)
23     # tstack = Stack.new(opts)
24     # layout(tstack, &blk)
25     #end
26
27     def radio(opts = {}, &blk)
28       opts.merge! :app => @app
29       Shoes::Radio.new(self, opts, blk)
30     end
31
32     def list_box(opts = {}, &blk)
33       opts.merge! :app => @app
34       Shoes::List_box.new(self, opts, blk)
35     end
36
37     def flow(opts = {}, &blk)
38       opts.merge! :app => @app
39       swt_flow = Shoes::Flow.new(self, opts, blk)
40     end
41
42     def button(text, opts={}, &blk)
43       opts.merge! :app => @app
44       button = Shoes::Button.new(self, text, opts, blk)
45       #@elements[button.to_s] = button
46       #button
47     end
48
49     # Creates an animation that runs the given block of code.
50     #
51     # @overload animate &blk
52     # @param [Proc] blk Code to run for each animation frame
```

What about GUI Applications?



Shoes

Past, present, future and introduction

Tobias Pfeiffer

@PragTob

pragtob.wordpress.com



No!

Isn't shoes dead?

What is shoes?



- GUI DSL/toolkit for Ruby
- Inspired by the web
- All ruby all the time
- Cross-platform and packaging
- Simple
- Feels like Ruby
- Many different implementations



I ♥ Shoes

Applications



Hackety Hack

Programs Samples

Check out the Hackety Hack Tour to get started!

en! %
7-03
ngtone, I'd
one. :-)
: I now hav
ne-ass today
t
y keyboard
e Animator
ailsCasts
else wants
[Go to Tweet](#)
log in to
Quora.
and how
cIn7olVT
ing? Tips
-07-03

Let's Dance

20:44:20 +0200 [Go to Tweet](#)

Tobias Pfeiffer: @nbeet ThoughtWorks :-| 2012-07-03 20:41:26



Quick History Walkthrough

Who knows who this guy
is?



why the lucky stiff

<https://en.wikipedia.org/wiki/File:Whytheluckystiff.jpg>

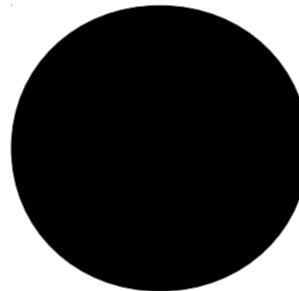
New maintainer



Steve Klabnik

http://25.media.tumblr.com/tumblr_lxlhv9OUO01qmjkado1_500.jpg

Many different colors of shoes



Shoes problems



- Many different implementations, but none fully working...
- “official” red shoes
 - rather a C-project and a ruby interpreter of its own
 - Packaging and installer of red shoes (partially) broken
 - unstableness
 - It isn't a gem
- There is no official “Shoes spec”

Shoes 4



- Concentrate efforts on one implementation
- Make the gui backend pluggable
- TDD implementation
- Based on brown shoes
- JRuby and SWT as a start (Qt?)

The pluggable backend



The Shoes DSL

The GUI backend

SWT

Qt

GTK

...



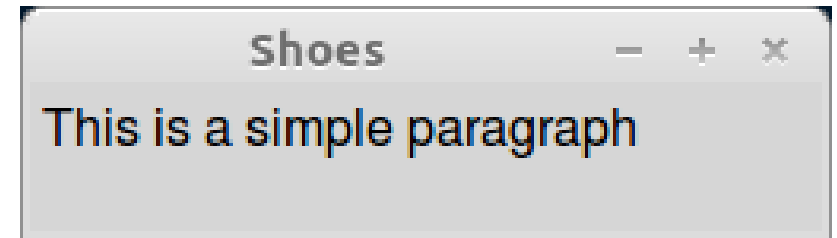
The Shoes DSL

(with huge thanks to [Nobody Knows Shoes](#))

Para



```
Shoes.app do
  para "This is a simple paragraph"
end
```



Flow

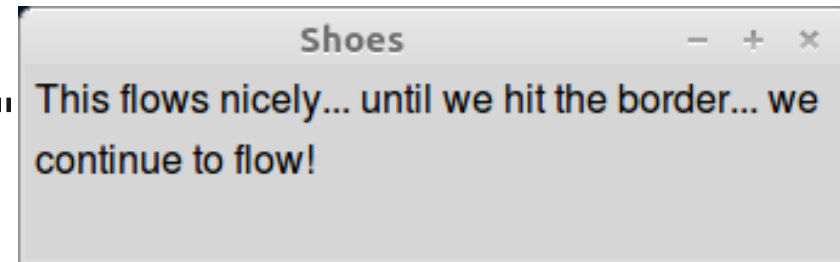


A **flow**, on the other hand, is more like a box of matches. As you fill it up, the matches squeeze in side-to-side. Eventually, it'll fill upward. But only after side-to-side. Sardines, right?

Flow



```
Shoes.app do
  flow do
    para "This flows nicely..."
    para "until we hit the border..."
    para "we continue to flow!"
  end
end
```



Stack

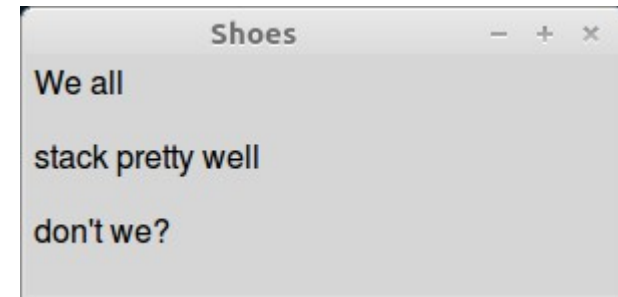


A stack. Let's say: of dominoes. A stack of dominoes. Banded together with a rubber band. Add more, the stack grows upward.

Stack



```
Shoes.app do
  stack do
    para "We all"
    para "stack pretty well"
    para "don't we?"
  end
end
```



More complex layouts



And more complex mixtures. The main Shoes window is itself a flow. Mixing stacks and flows builds columns.

More complex layouts



```
Shoes.app do
```

```
  flow do title "A nice title" end
```

```
  flow do
```

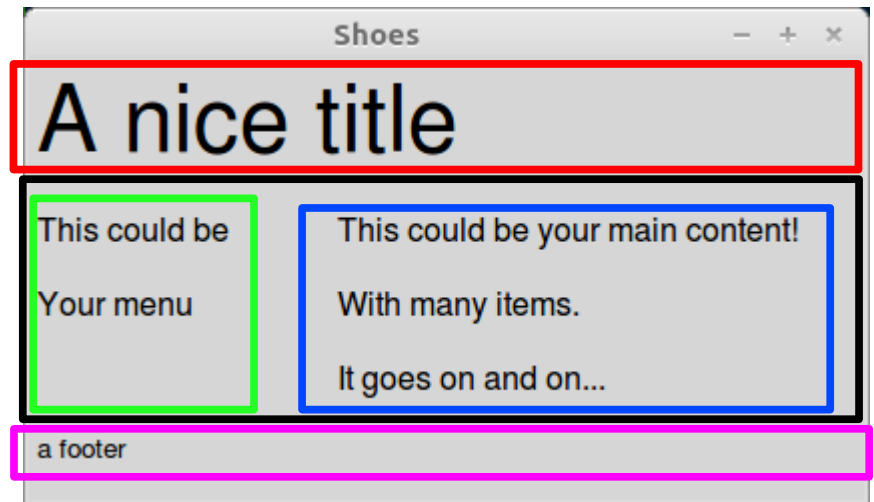
```
    stack width: 150 do  
      para "This could be"  
      para "Your menu"  
    end
```

```
    stack width: -150 do  
      para "This could be your  
        main content!"  
      para "With many items."  
      para "It goes on and on..."  
    end
```

```
  end
```

```
  flow do inscription "a footer" end
```

```
end
```



Button



```
Shoes.app do
  button "Click me!" do
    alert "Good job!"
  end
end
```



Image



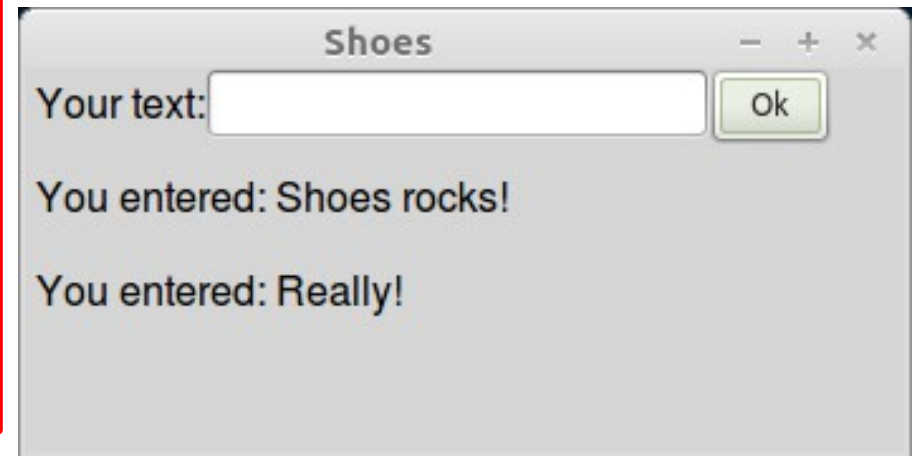
```
Shoes.app width: 270, height: 610 do
  image "http://www.rug-b.de/Images/
  Rug-b-ruby-user-grouperberlin.jpg",
  click: "http://www.rug-b.de/"
end
```



Editline & Slots



```
Shoes.app do
  para "Your text:"
  @edit = edit line
  button "Ok" do
    text = @edit.text
    @slot.append do
      para "You entered:#{text}"
    end
    @edit.text = ""
  end
  @slot = stack do ; end
end
```



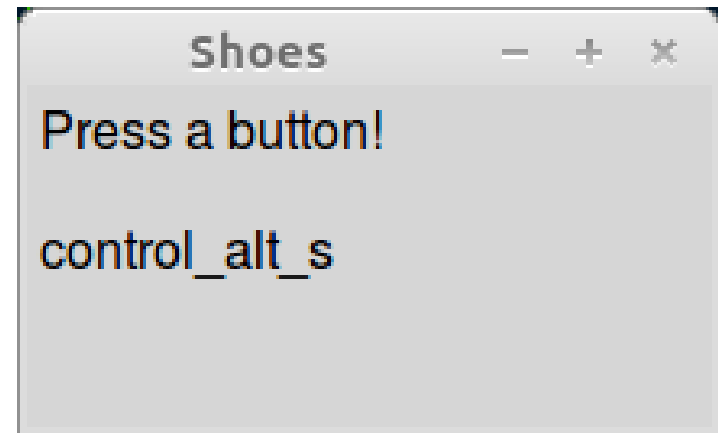
Keypresses



```
Shoes.app do
  para "Press a button!"
  @slot = stack do ; end
```

```
  keypress do |key|
    @slot.clear do para key end
  end
```

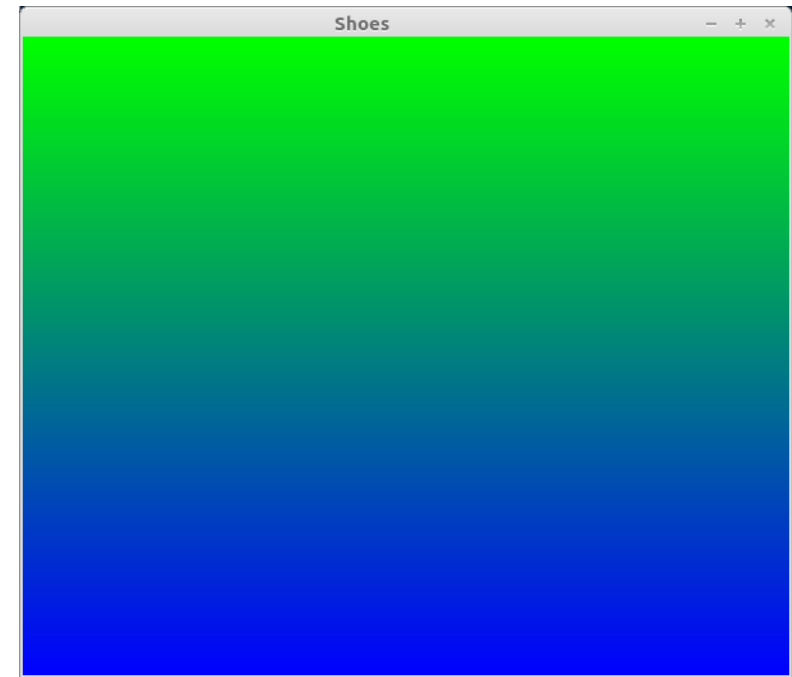
```
end
```



Background & gradient



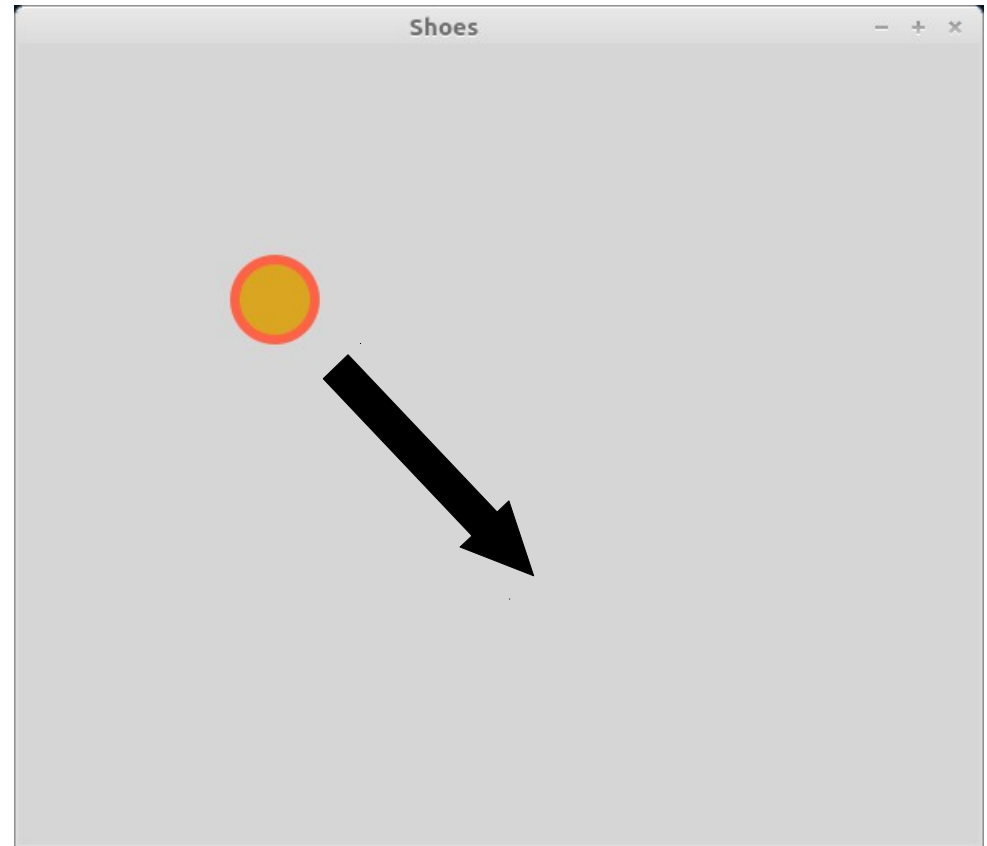
```
Shoes.app do  
  background gradient("#0F0", "#00F")  
end
```



Animate



```
Shoes.app do
  fill goldenrod
  stroke tomato
  strokewidth 6
  @oval = oval 0, 0, 50
  animate 24 do |frame|
    @oval.move frame, frame
  end
end
```



Using gems with red shoes



```
Shoes.setup do
  gem 'twitter'
  gem 'oauth'
  gem 'launchy'
end
```

```
require 'launchy'
(...)
```

```
Shoes.app do
  (...)
end
```





Now you know Shoes!



Wrap-up

Wrap-up

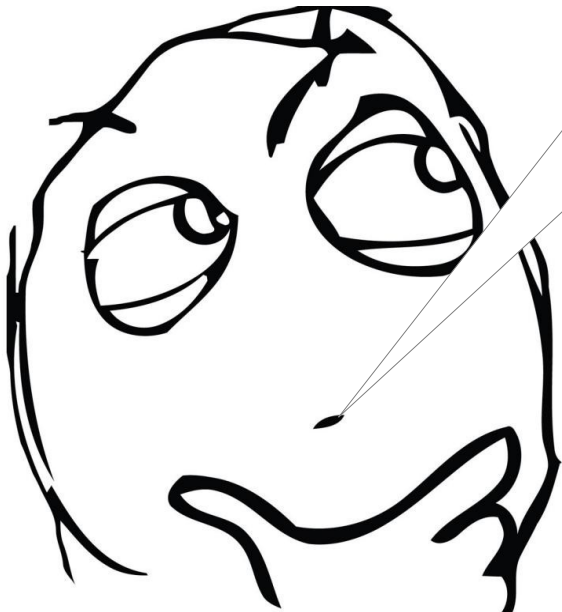


- Shoes is a nice Ruby GUI-toolkit
- There are many colors of Shoes
- Super cool for small applications
- Pro-Tip: Open the red shoes console with
Alt + /



Wait a second!
How do I test
my Shoes apps?

Only test your models
until we've got a Shoes
mocking backend!



Further information



- <http://shoesrb.com>
- <http://shoesrb.com/manual/Hello.html>
- shoes@librelist.com
- #shoes on freenode
- @shooooesrb
- <http://github.com/shoes/shoes4> ← We could use some help!
- [The Return of Shoes](#) by Steve Klabnik
- [Nobody Knows Shoes](#) by `_why`

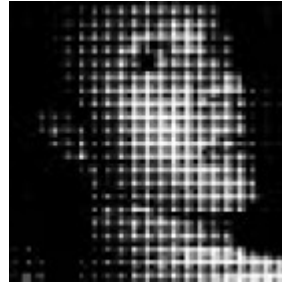
Thank you Team shoes!



steveklabnik



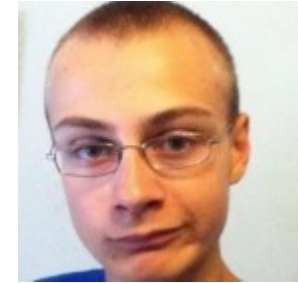
ashbb



wasnotrice



pjfitzgibbons



jrgifford



mpapis



davorb



ccoupe



ljk



_why

And all the ones I forgot...

(All pictures and names taken from github)



Shoes on!

Tobias Pfeiffer
@PragTob