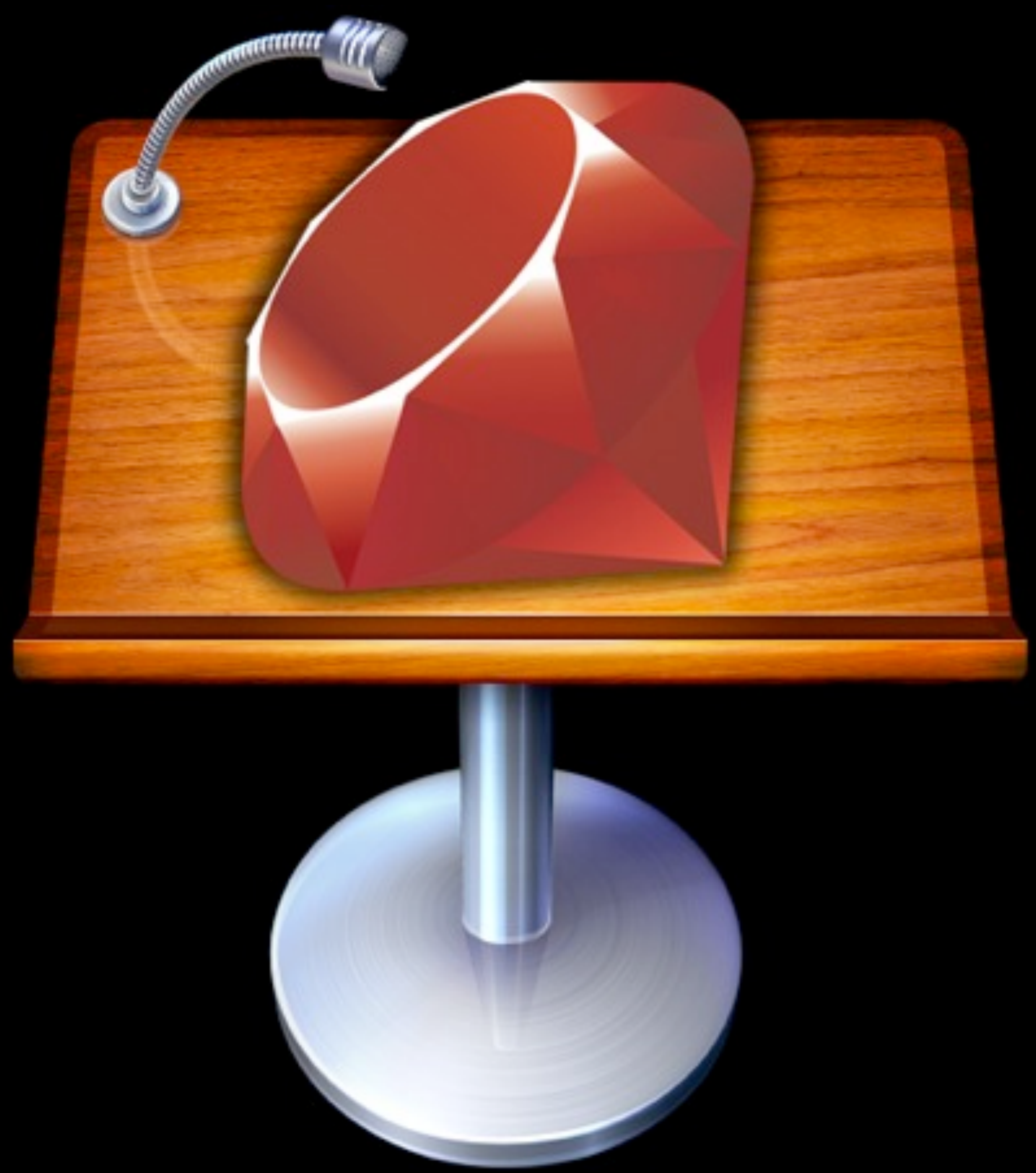
Basking in the

limelight

by Micah Martin
8th Light, Inc.

8th light

Limelight is....



(Ruby on the Desktop)

Ideas From The Web

Ideas From The Web

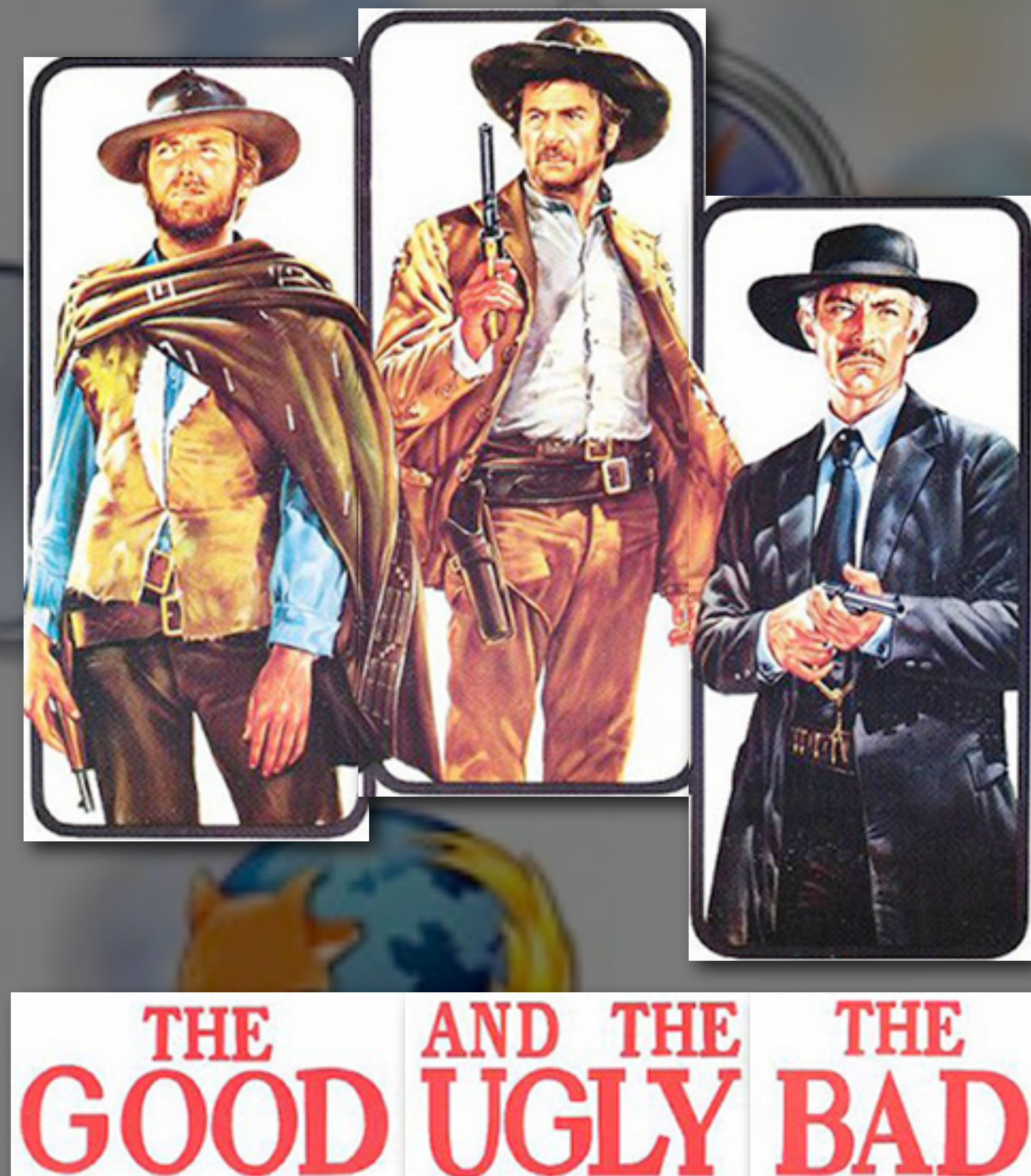
THE
GOOD

Ideas From The Web

THE GOOD

THE BAD

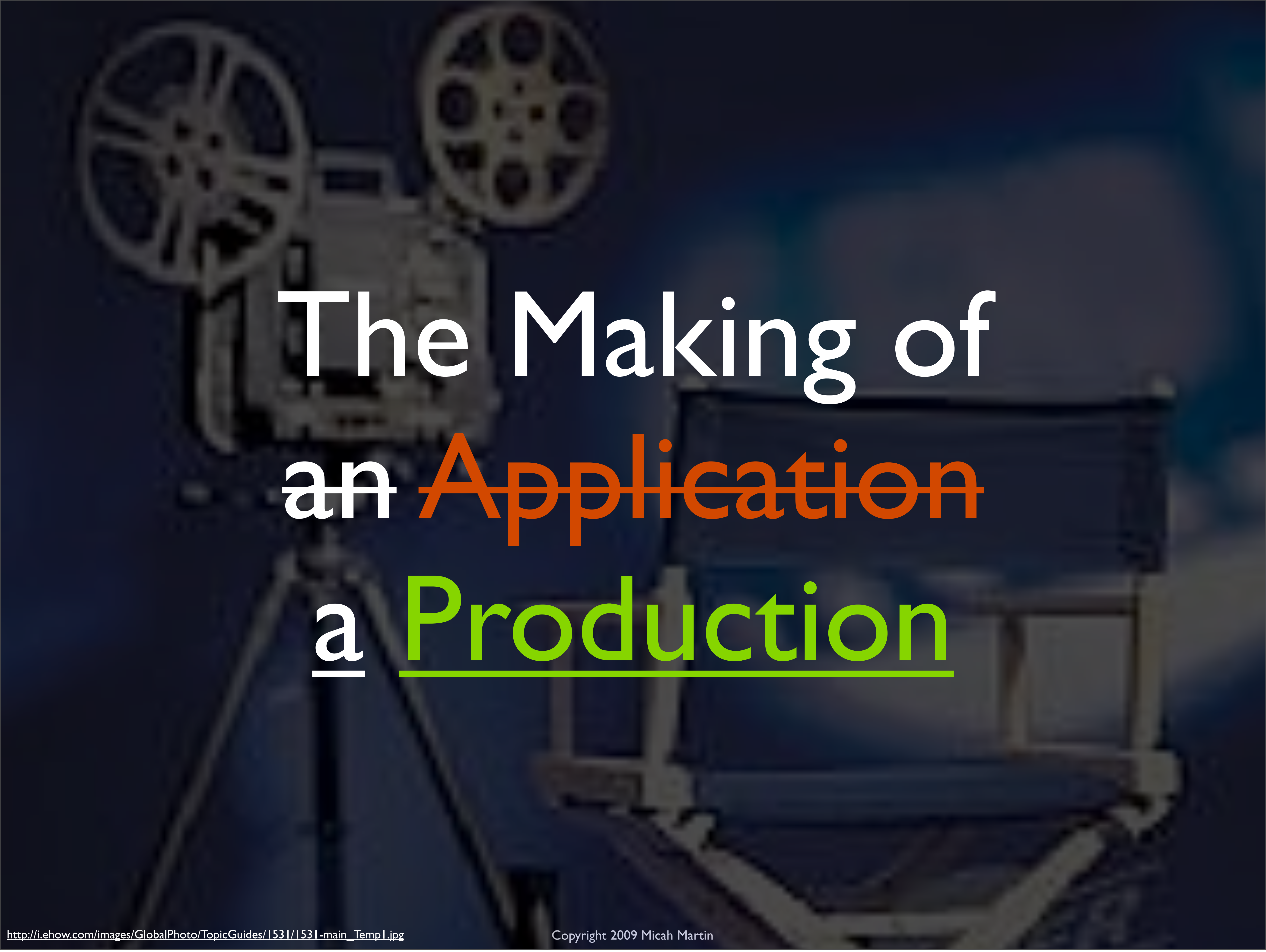
Ideas From The Web



Theater Metaphor

http://jacatering.com/images/Theater.jpg


Macintosh HD
Desktop



The Making of ~~an Application~~ a Production

http://i.ehow.com/images/GlobalPhoto/TopicGuides/1531/1531-main_Temp1.jpg

Setup

1. Install JRuby
2. Install Limelight Gem
    - jruby -S gem install gemcutter
    - jruby -S gem tumble
    - jruby -S gem install limelight

JRuby syntax to execute gem executable

limelight command to generate production files

Name of production to create

```
$ jruby -S limelight create production example
     creating directory:  ./example
     creating file:       ./example/production.rb
     creating file:       ./example/stages.rb
     creating file:       ./example/styles.rb
     creating directory:  ./example/spec
     creating file:       ./example/spec/spec_helper.rb
     creating directory:  ./example/default_scene
     creating file:       ./example/default_scene/props.rb
     creating file:       ./example/default_scene/styles.rb
     creating directory:  ./example/default_scene/players
     creating directory:  ./example/spec/default_scene
     creating file:       ./example/spec/default_scene/default_scene_spec.rb
```

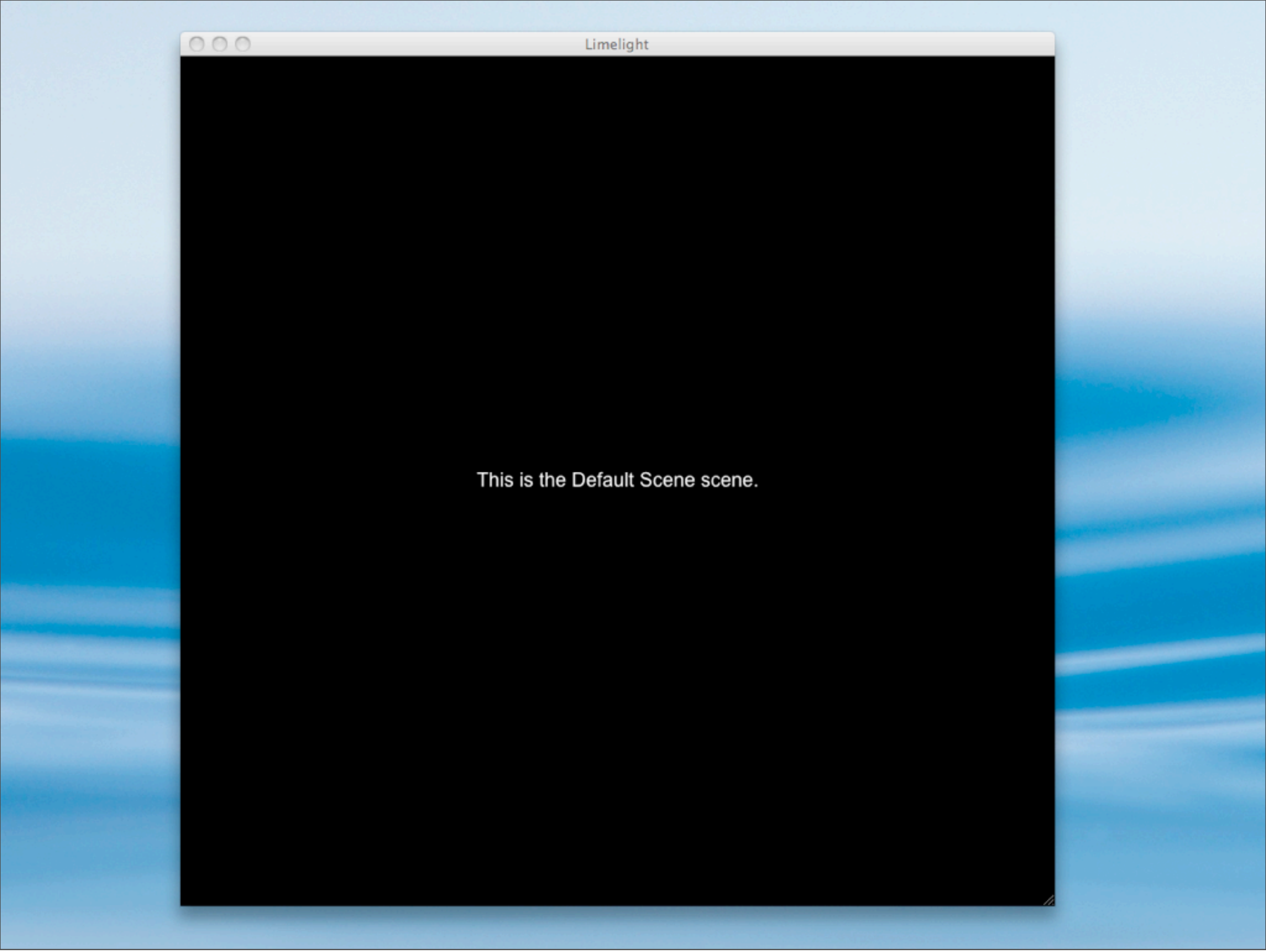
output from command

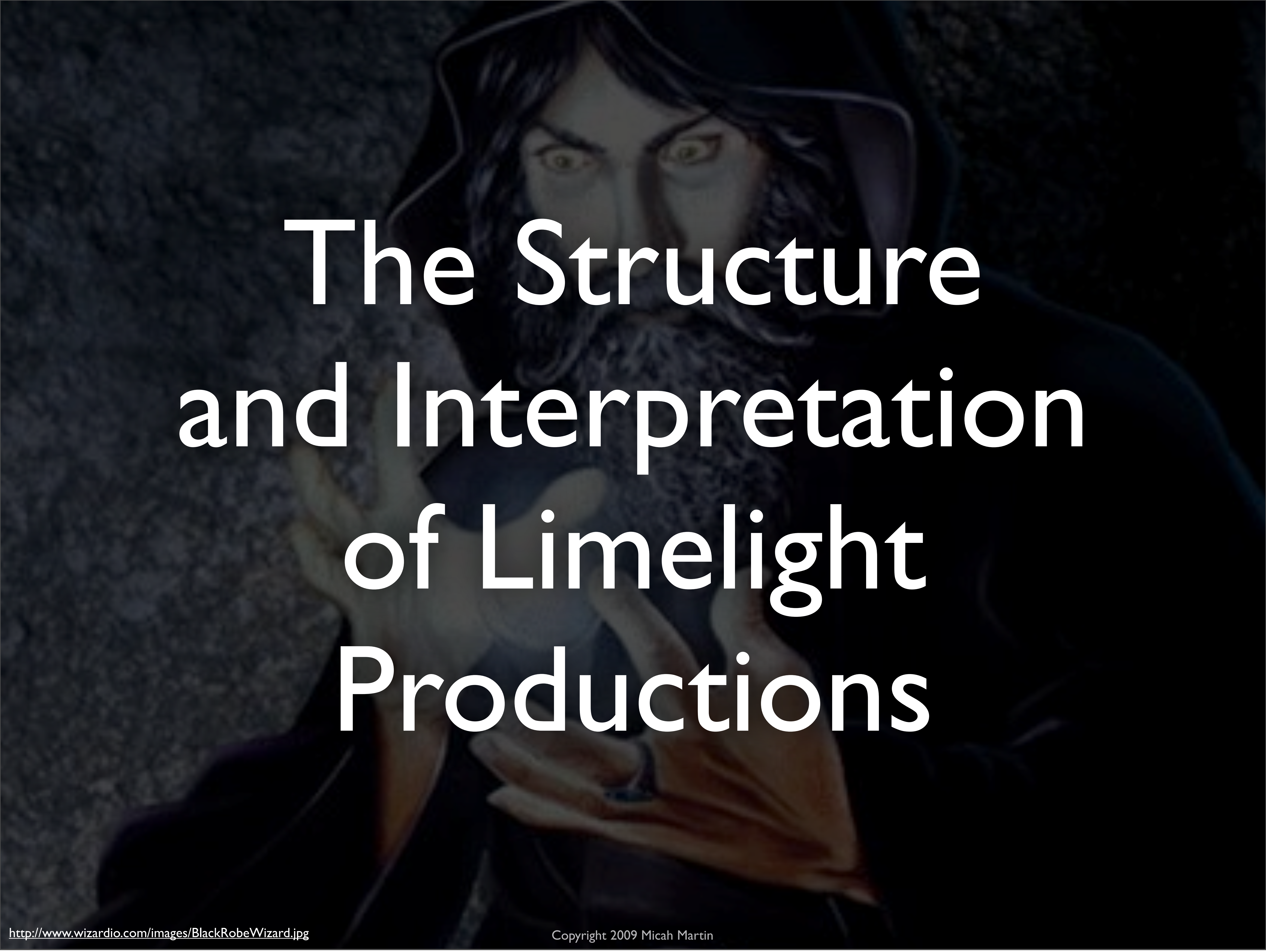
Opening a Production

```
1 $ jruby -S limelight open example
```

limelight command to generate production files

Name of production to create

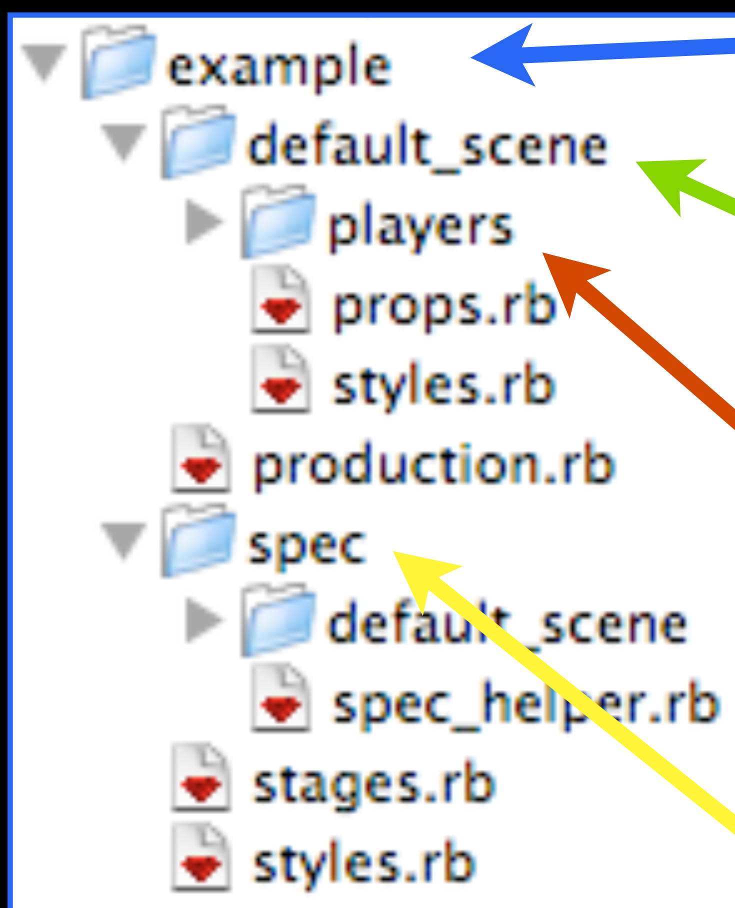

Limelight
This is the Default Scene scene.



The Structure and Interpretation of Limelight Productions

http://www.wizardio.com/images/BlackRobeWizard.jpg

Directories



production root
(named after production)

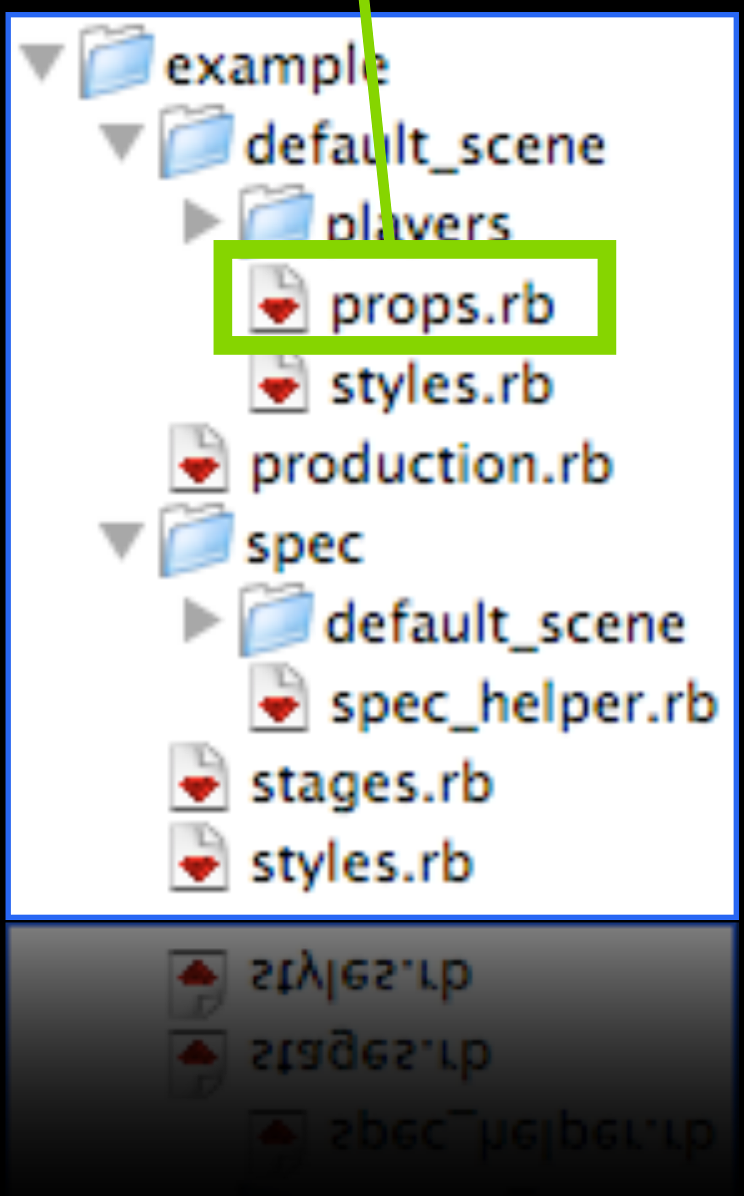
a scene
(named "default_scene")

players directory
(for the scene)

specs
(you do TDD right?)

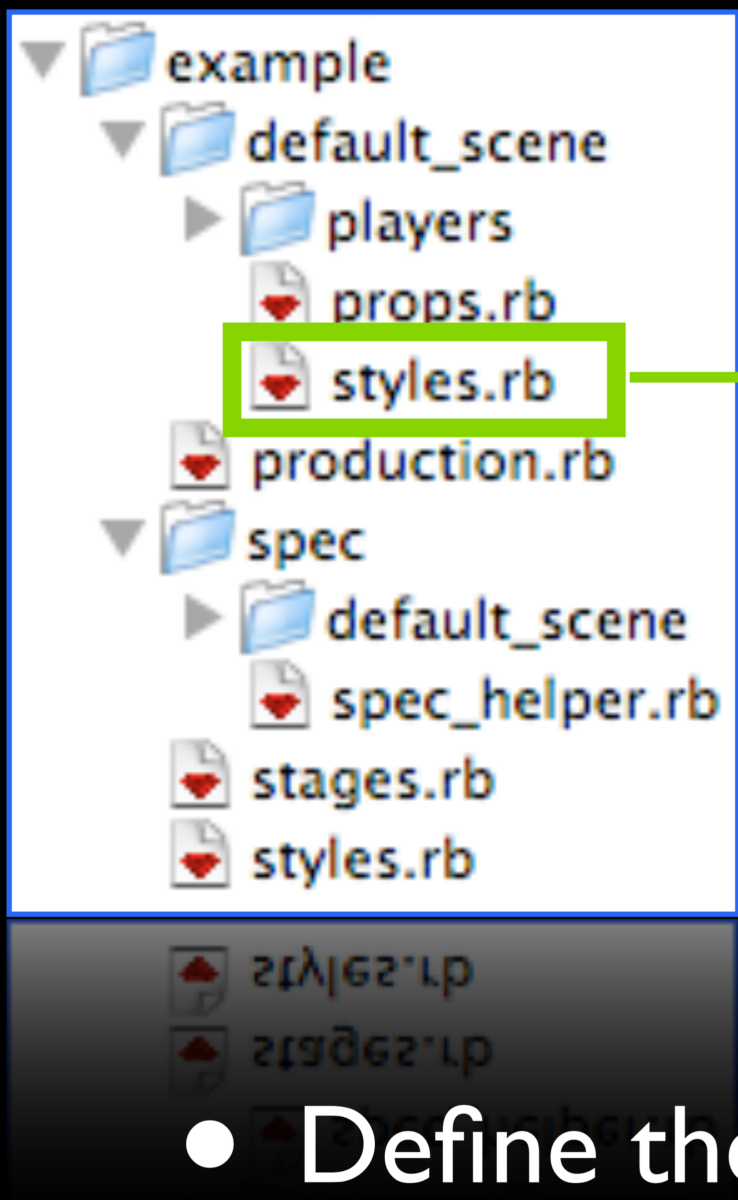
props.rb / Prop DSL

```ruby
root :text => "This is the Default Scene scene."
```



- Define the Structure of a Scene
- All Ruby Code
- DSL to build tree structure of props
- Arbitrary Prop names

styles.rb / Styles DSL




```ruby
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```

- Define the Look and Feel of a Scene
- All Ruby Code
- key:value based DSL

Naming conventions

styles.rb

```
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```

Naming conventions

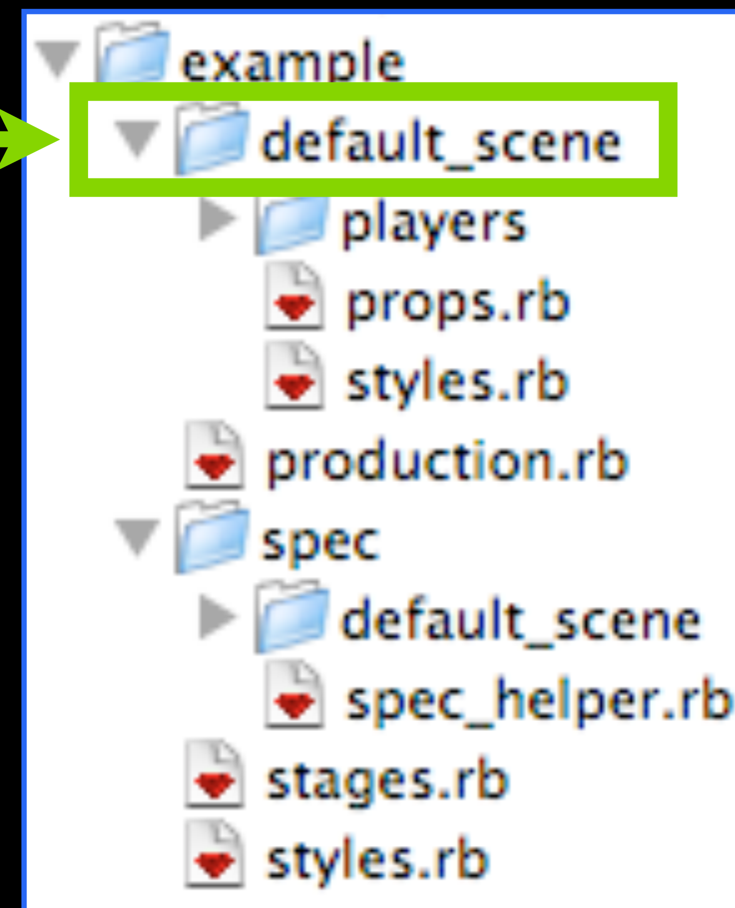
styles.rb

```ruby
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```

Naming conventions

styles.rb

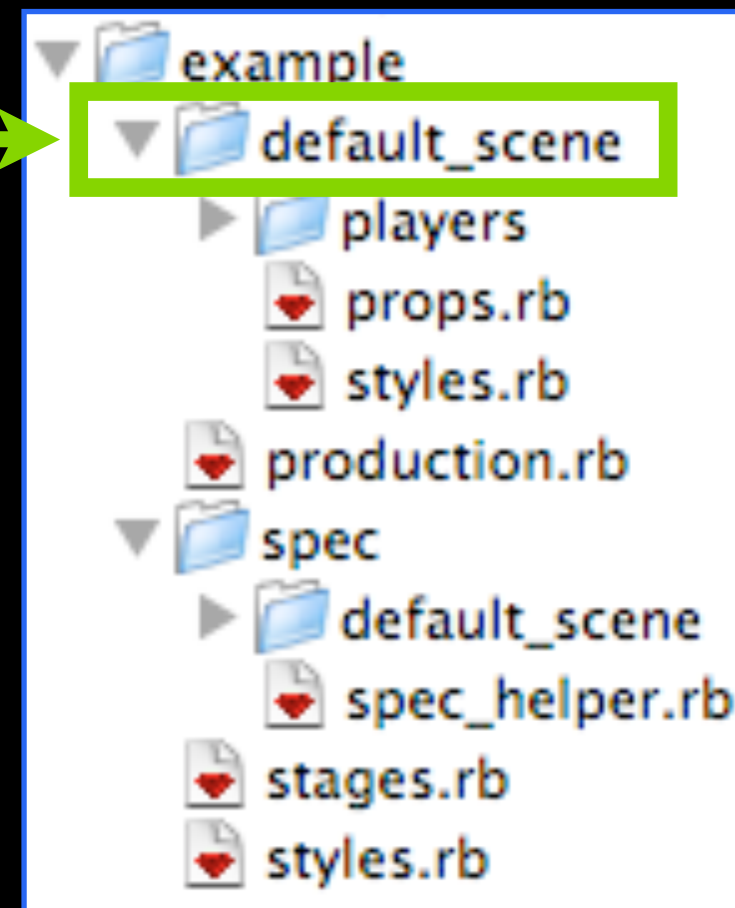
```
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```



Naming conventions

styles.rb

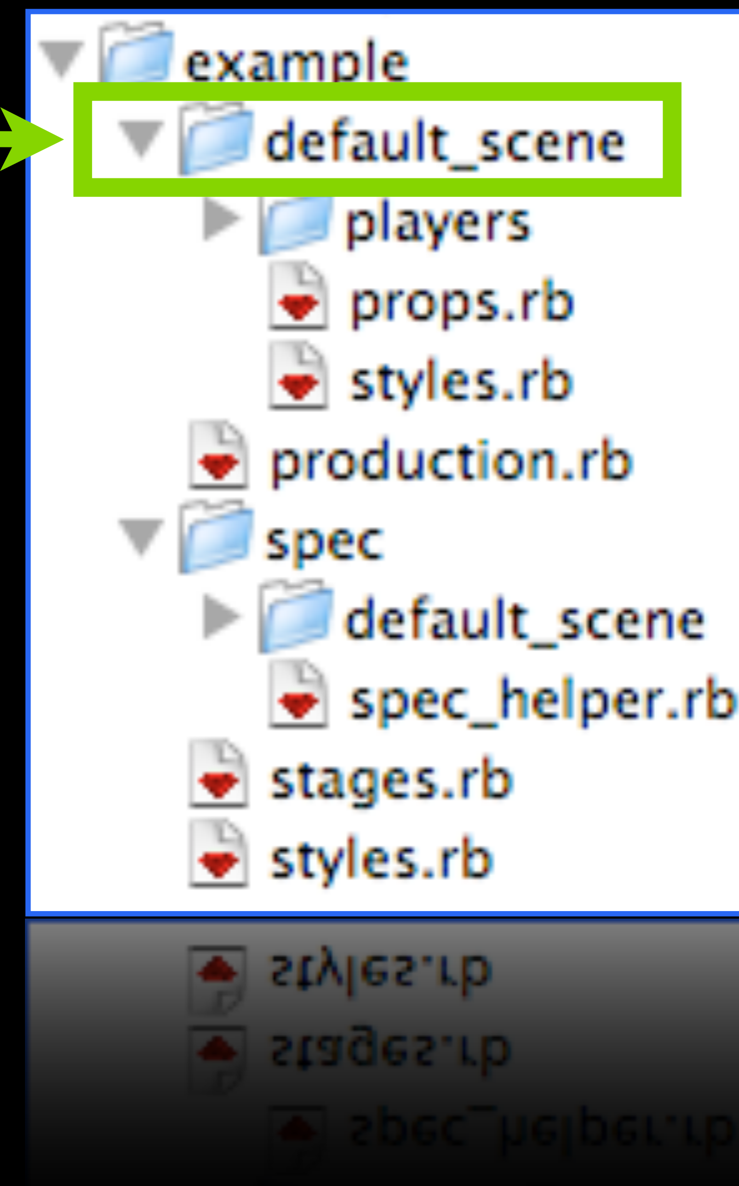
```
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```



Naming conventions

styles.rb

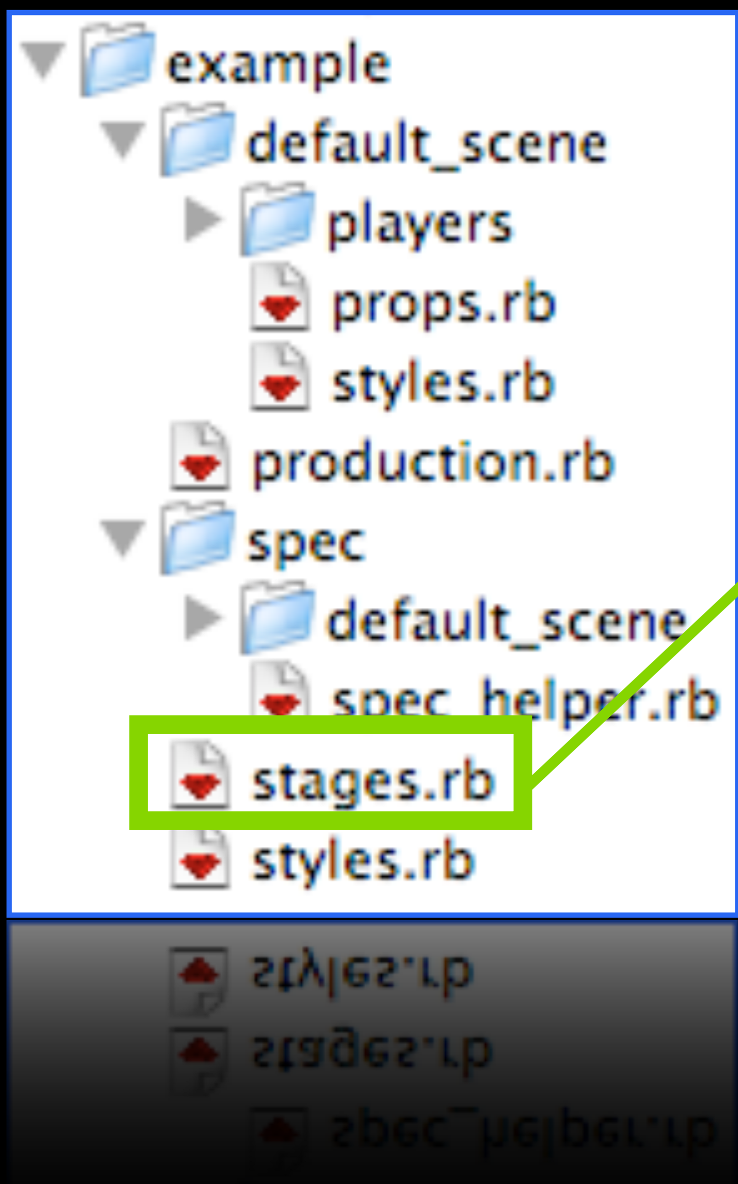
```
default_scene {
  background_color :black
  horizontal_alignment :center
  vertical_alignment :center
  width "100%"
  height "100%"
}

root {
  text_color :white
  font_size 18
}
```



props.rb

```
root :text => "This is the Default Scene scene."
```

stages.rb / Stage DSL




```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```

- Define and configure stages (windows)
- All Ruby Code
- key:value based DSL

Stage Name

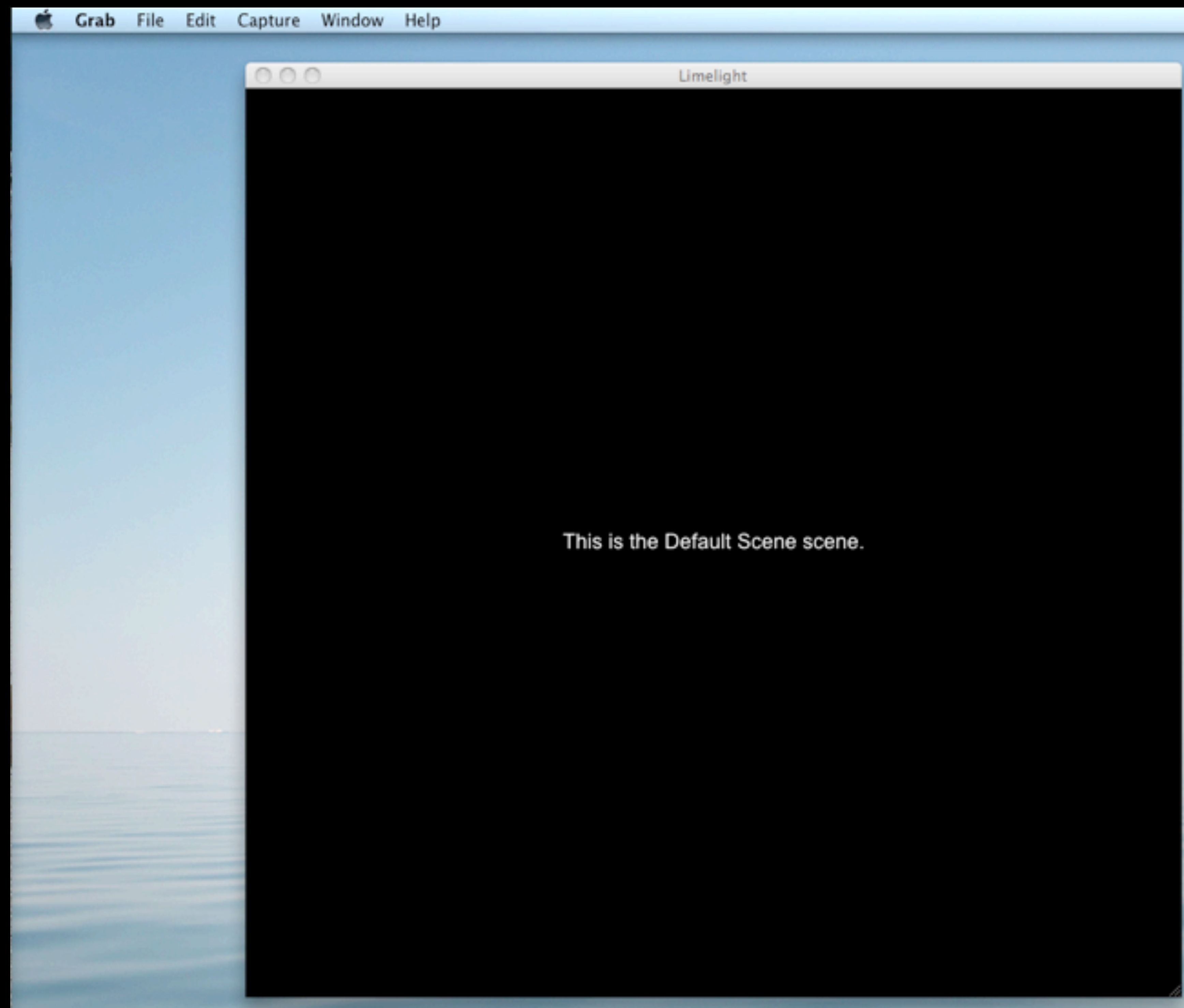
Default Scene

Title

Location

Size

```
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

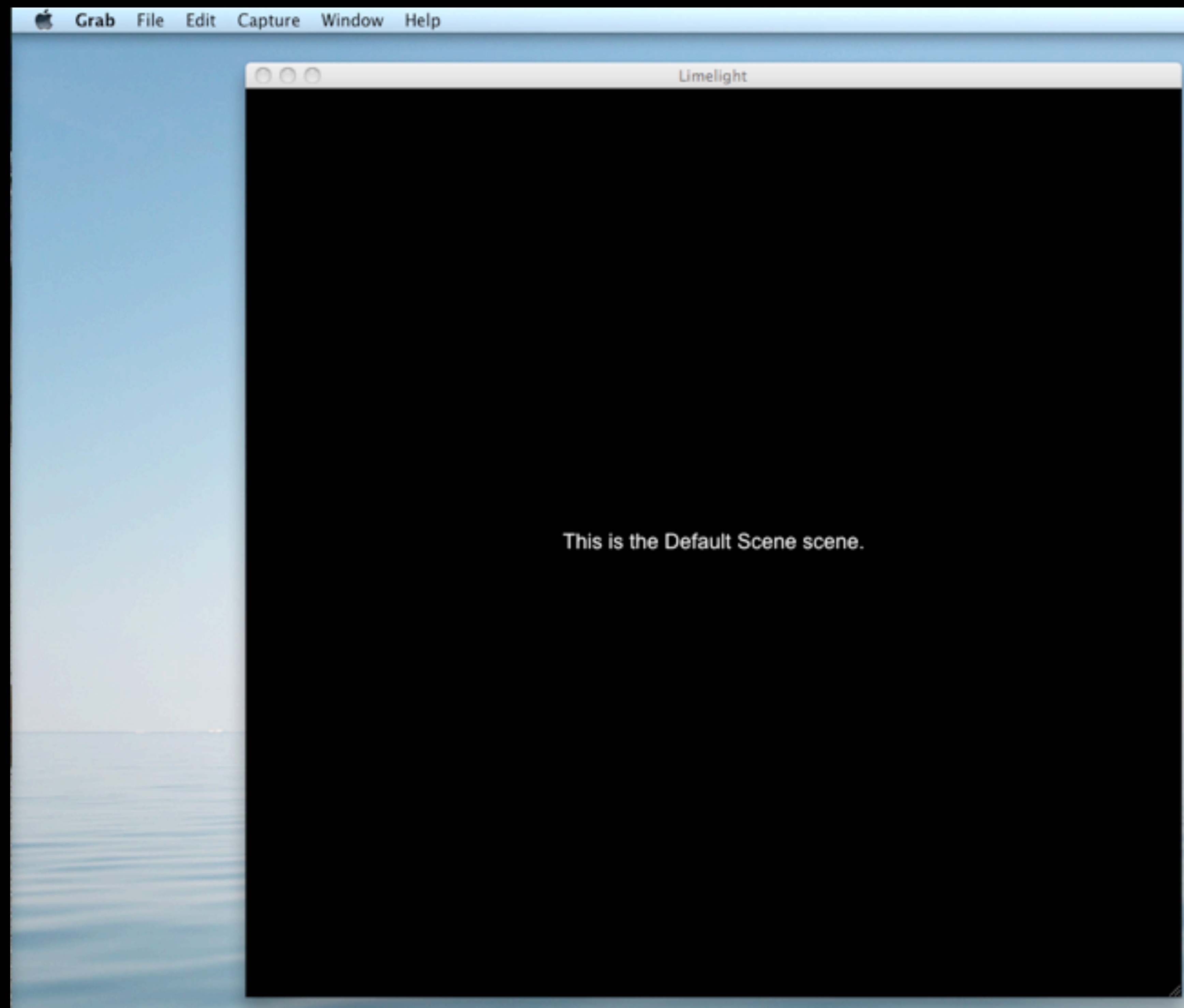
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

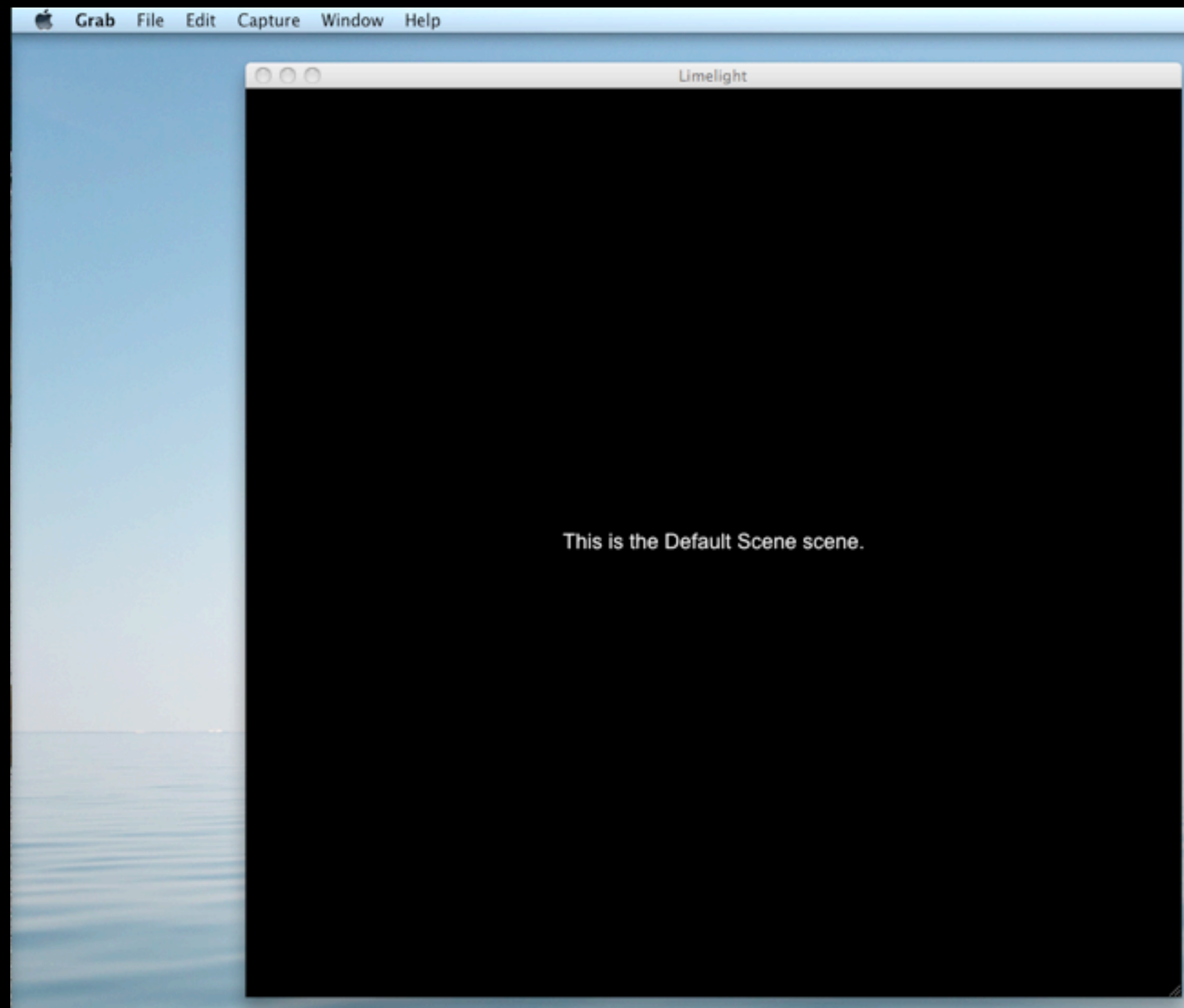
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

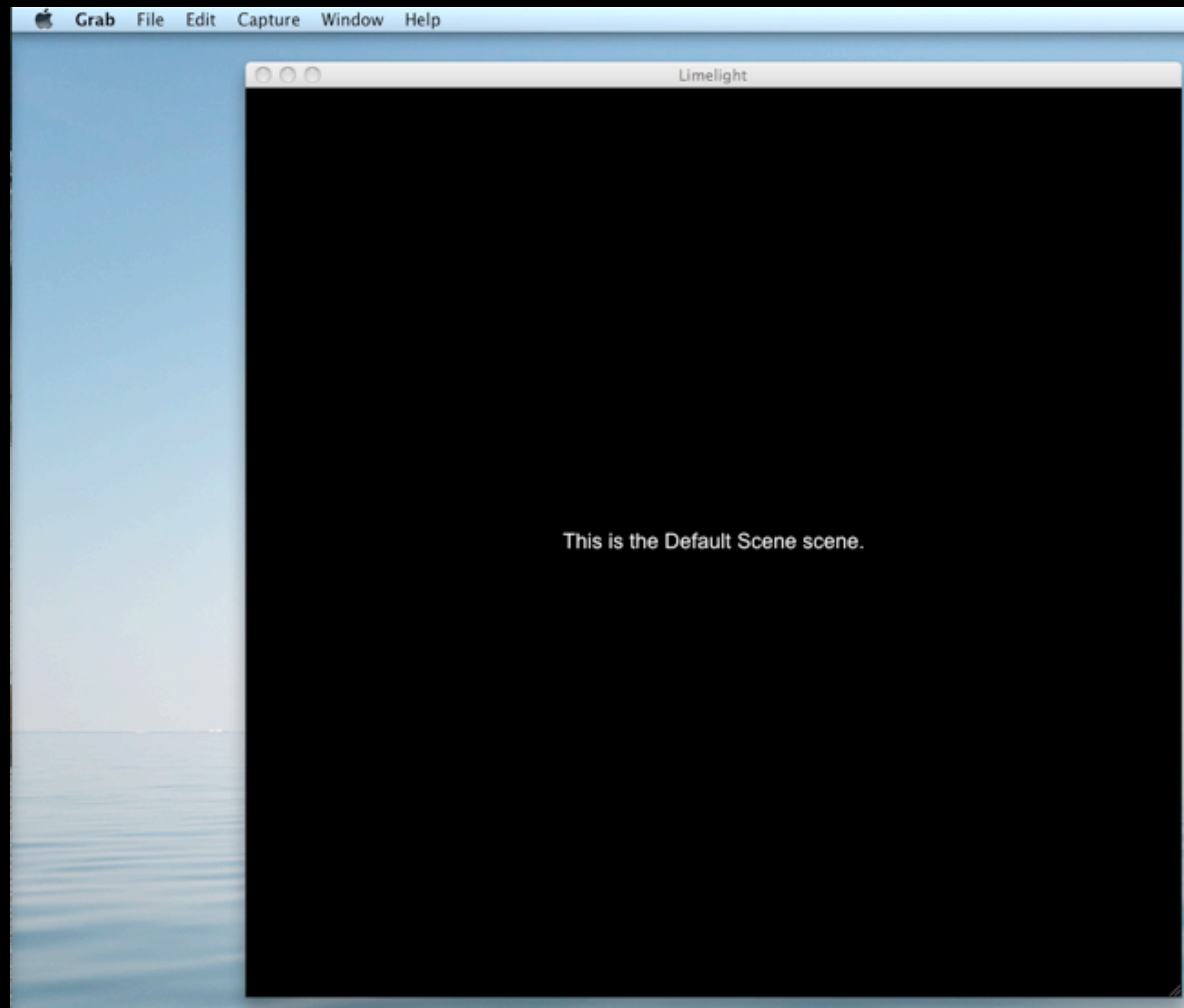
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

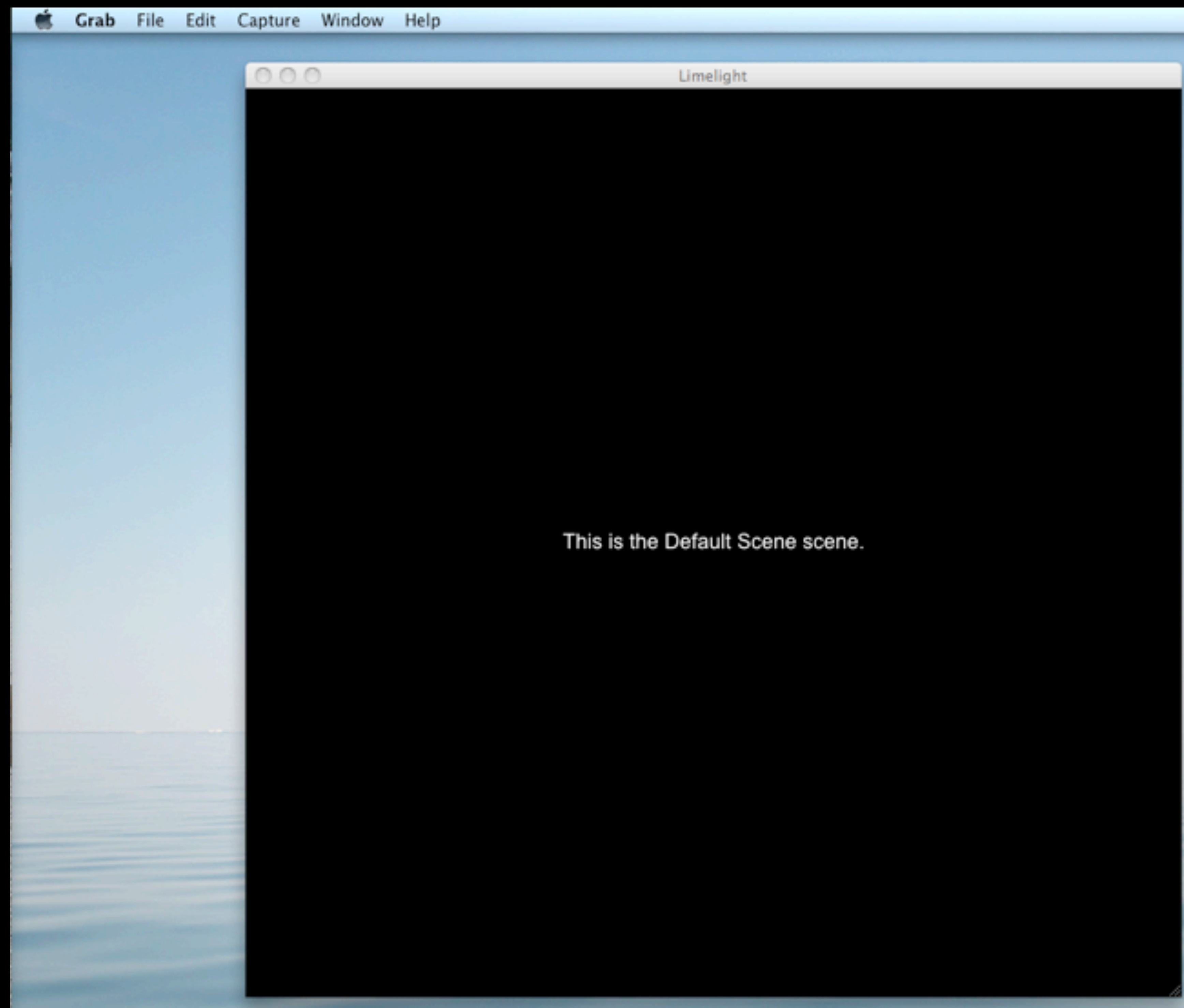
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

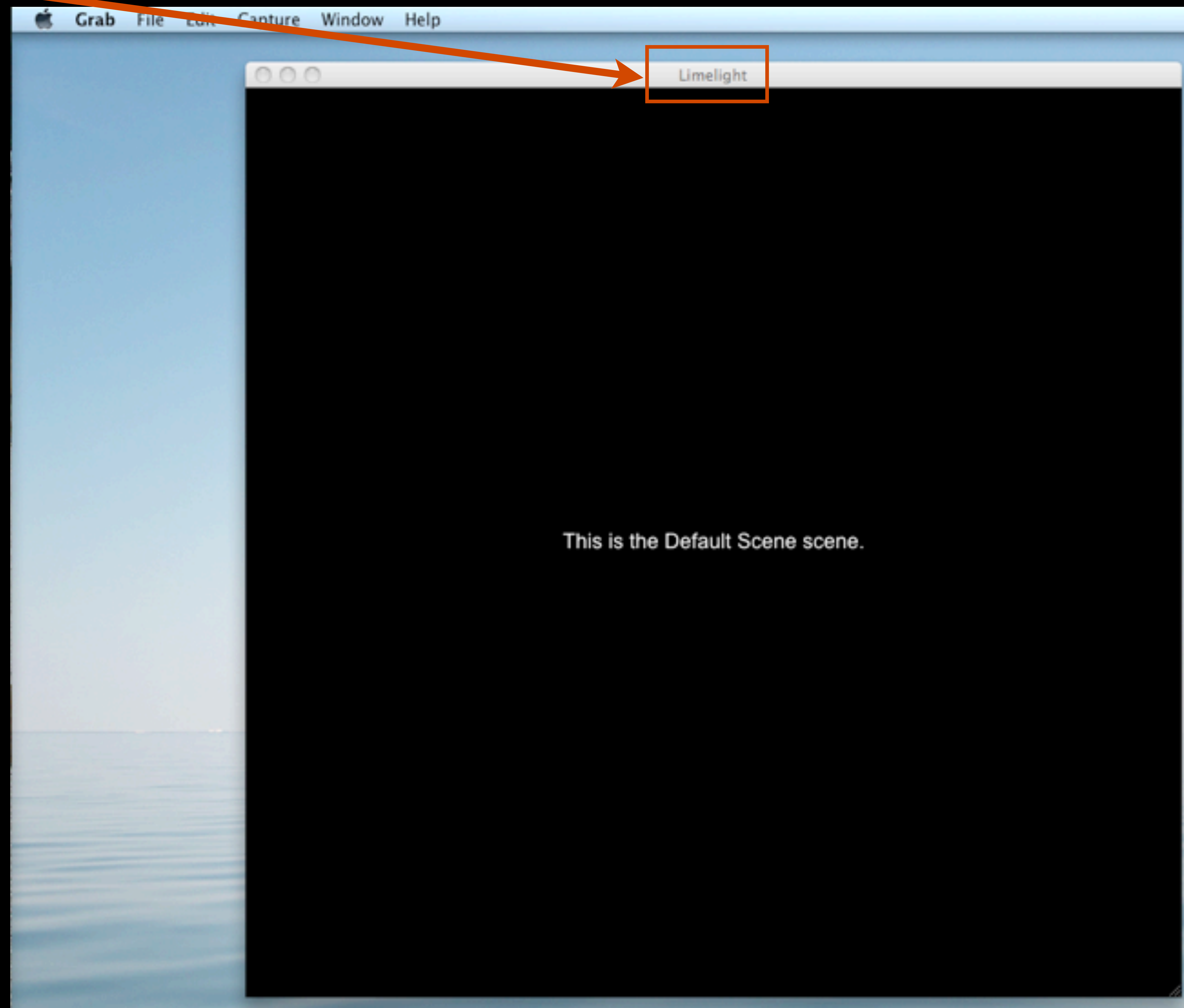
Default Scene

Title

Location

Size

```
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

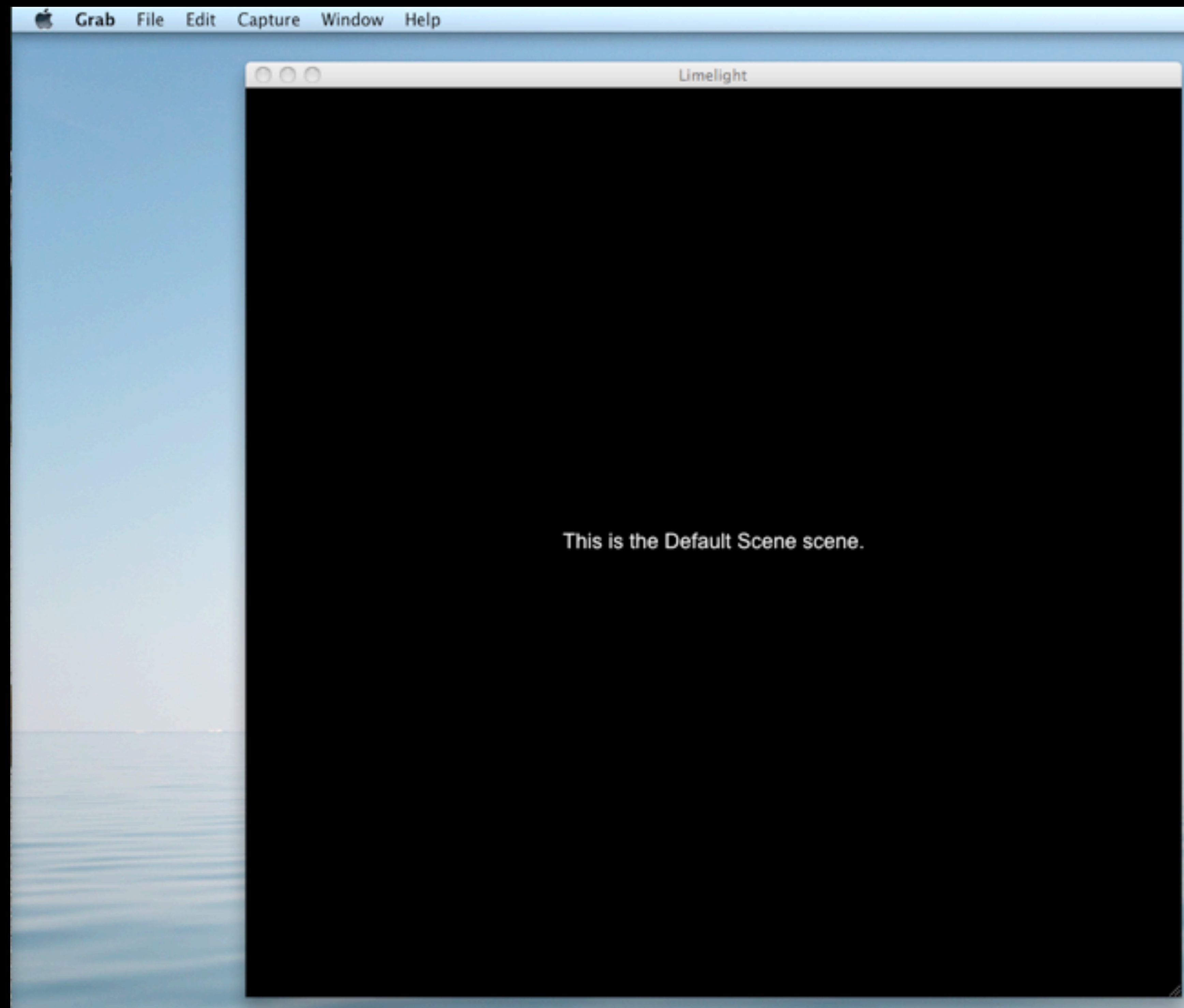
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

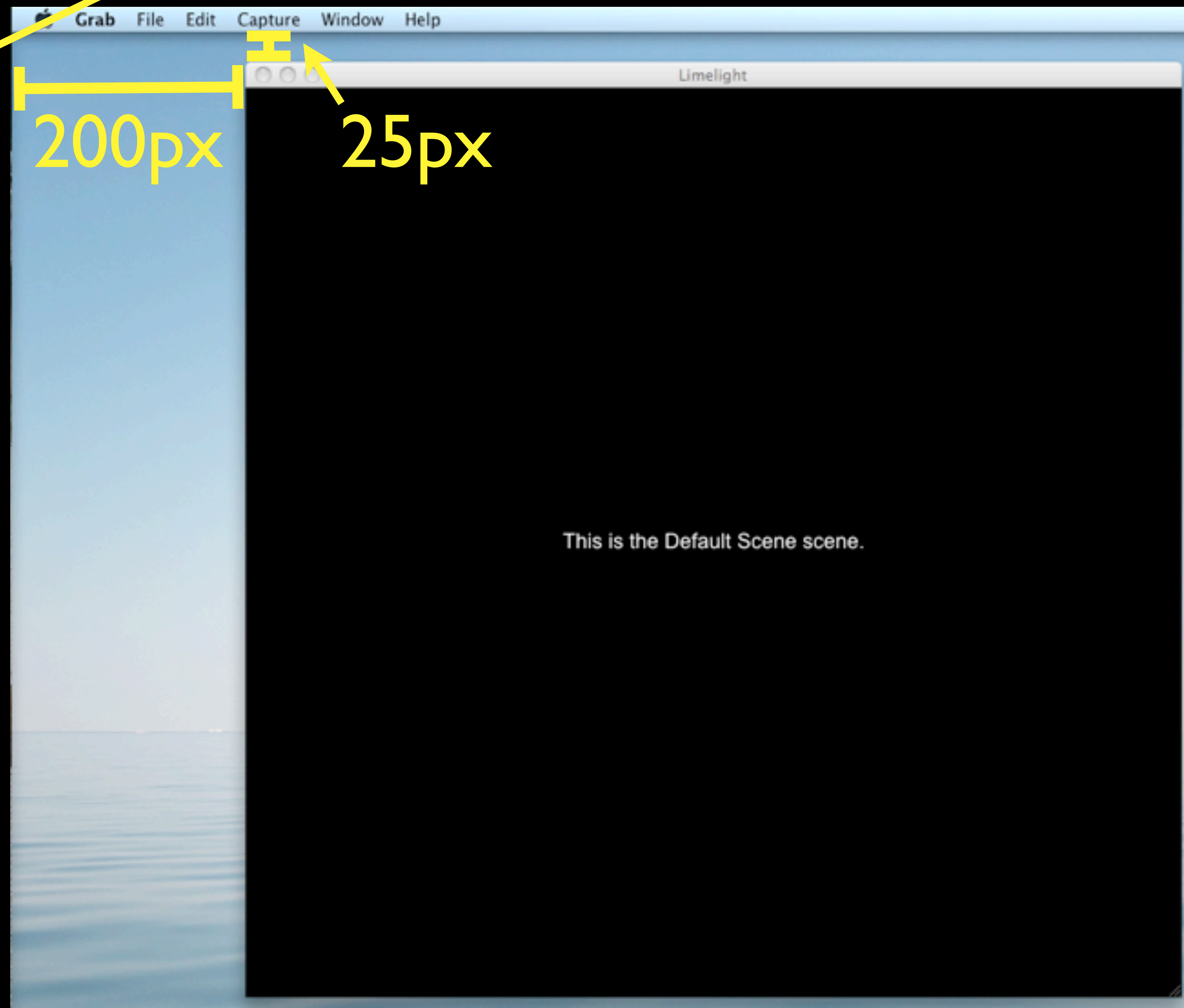
Default Scene

Title

Location

Size

```
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

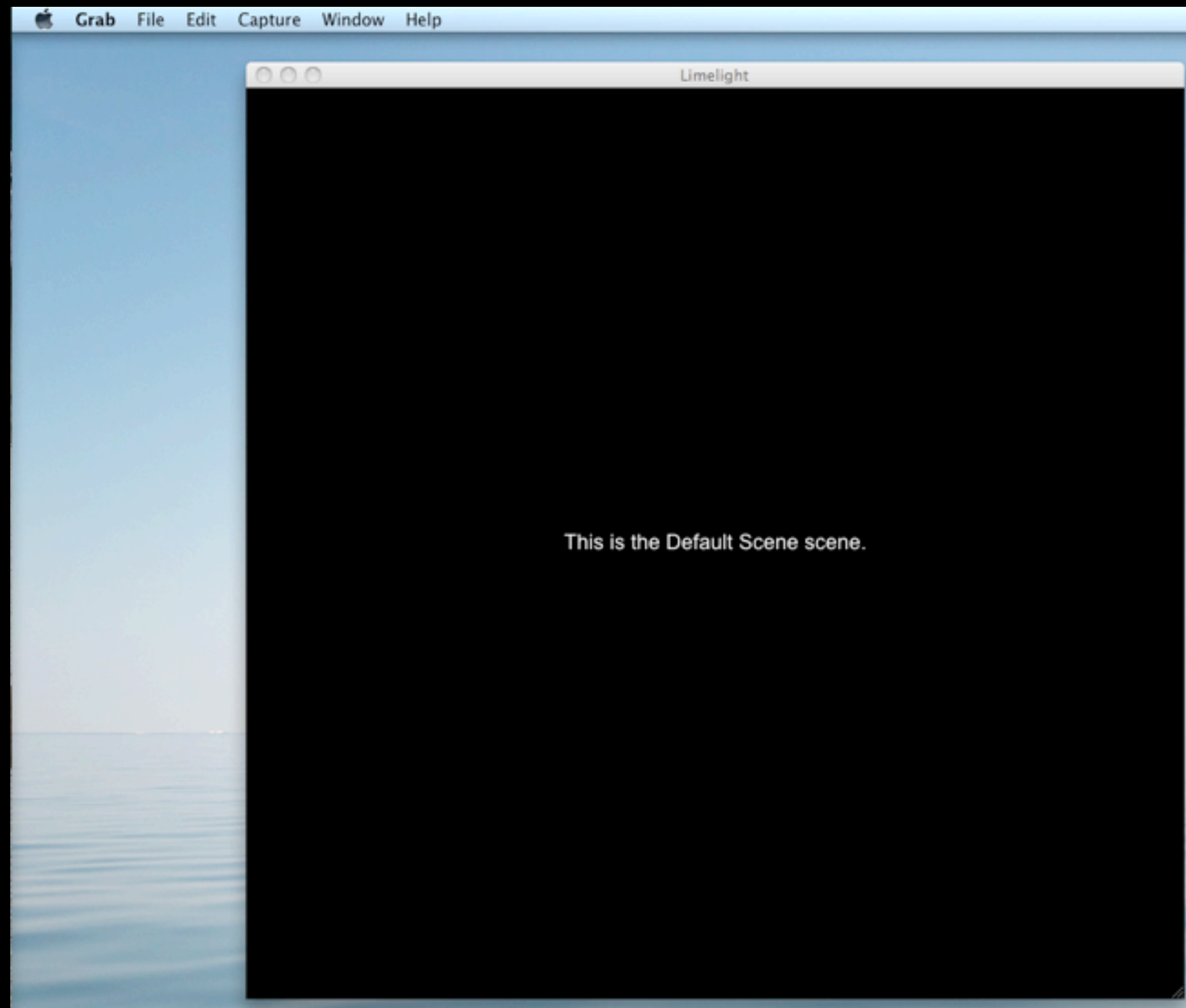
Default Scene

Title

Location

Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

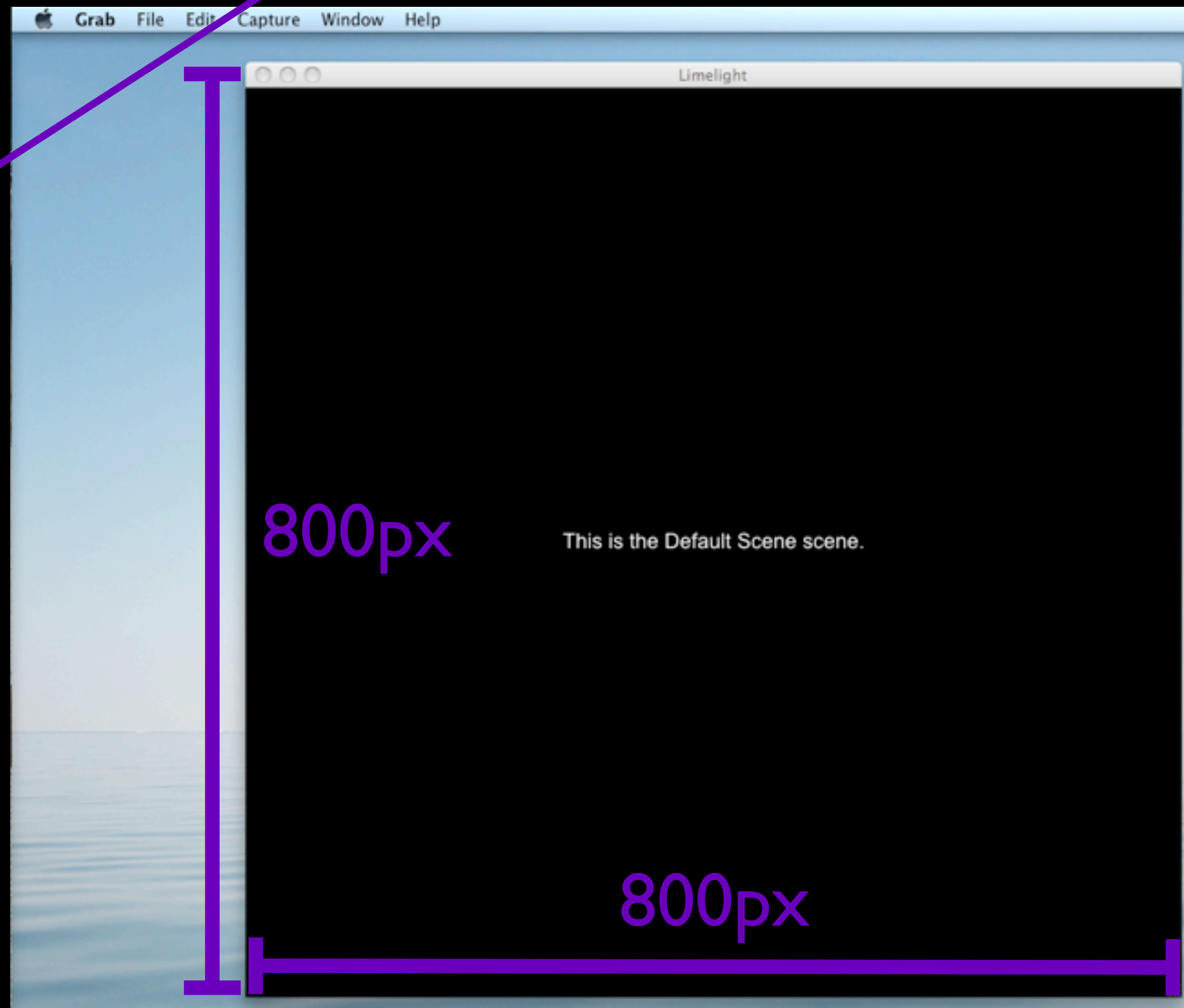
Default Scene

Title

Location

Size

```
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



Stage Name

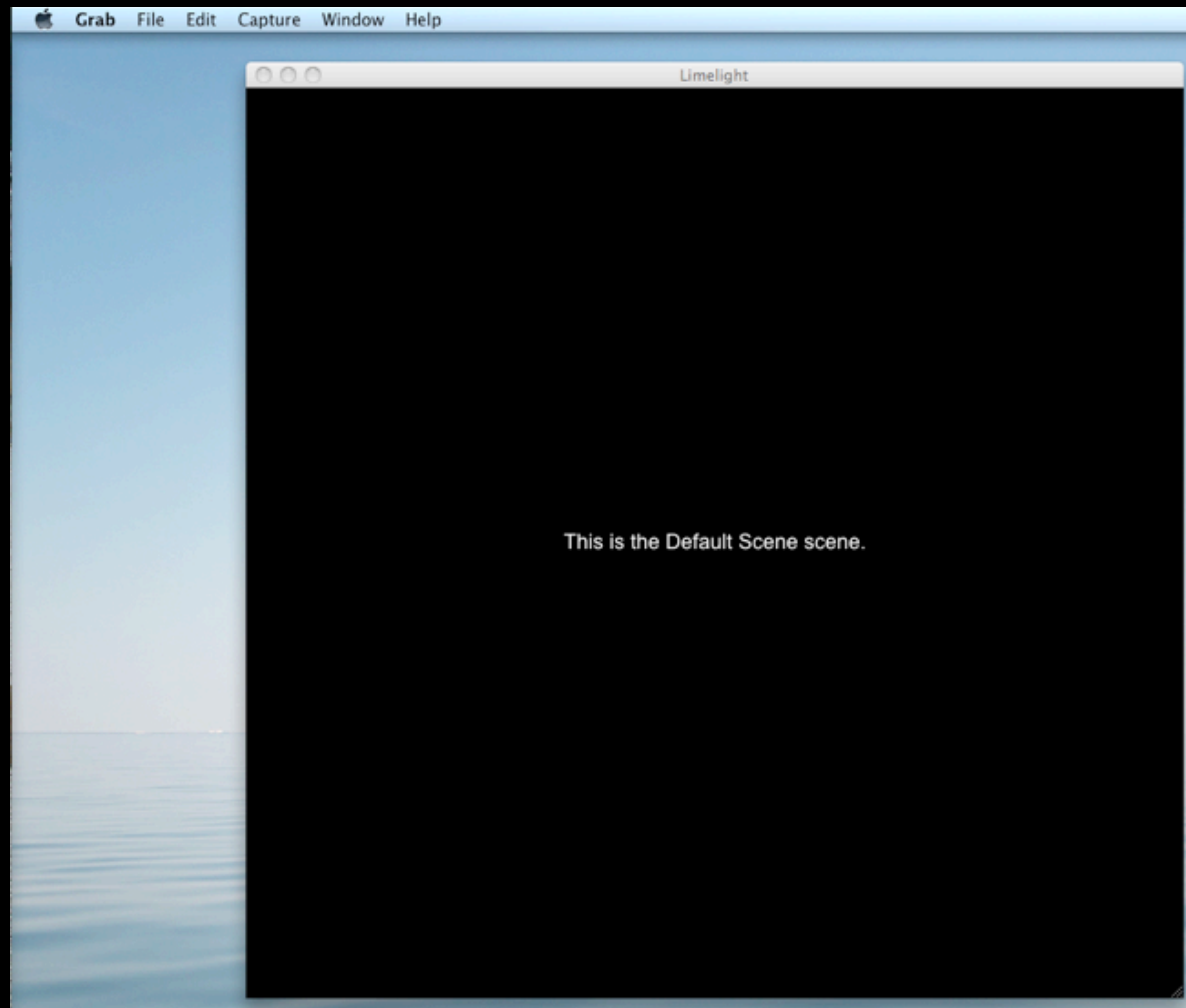
Default Scene

Title

Location

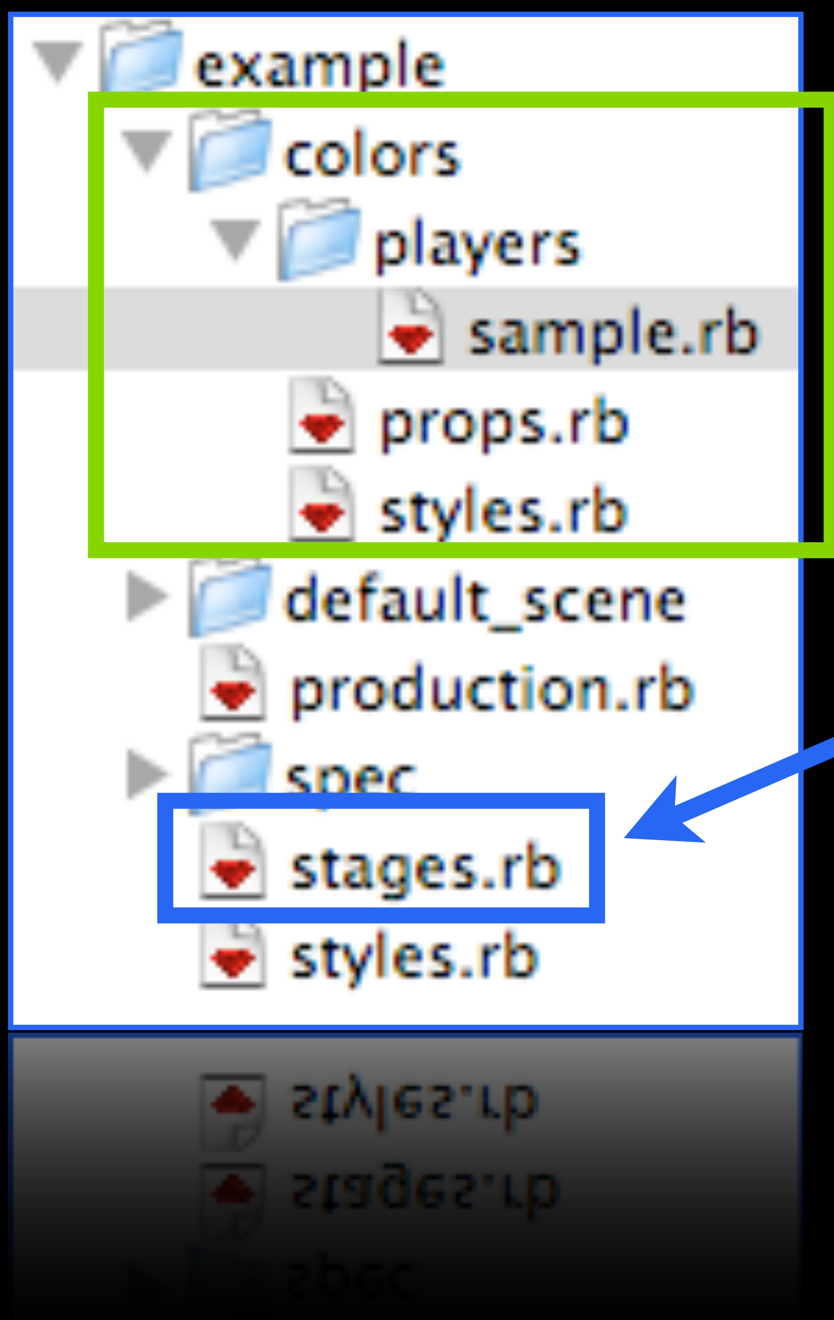
Size

```ruby
stage "default" do
  default_scene "default_scene"
  title "Limelight"
  location [200, 25]
  size [800, 800]
end
```



All the World's a Stage and We are Merely Players

Colors Scene



New Scene named "colors"

Changes to stages.rb

```
stage "colors" do
  default_scene "colors"
  title "Colors"
  location :center, :center
  size [700, 400]
end
```

Colors Code

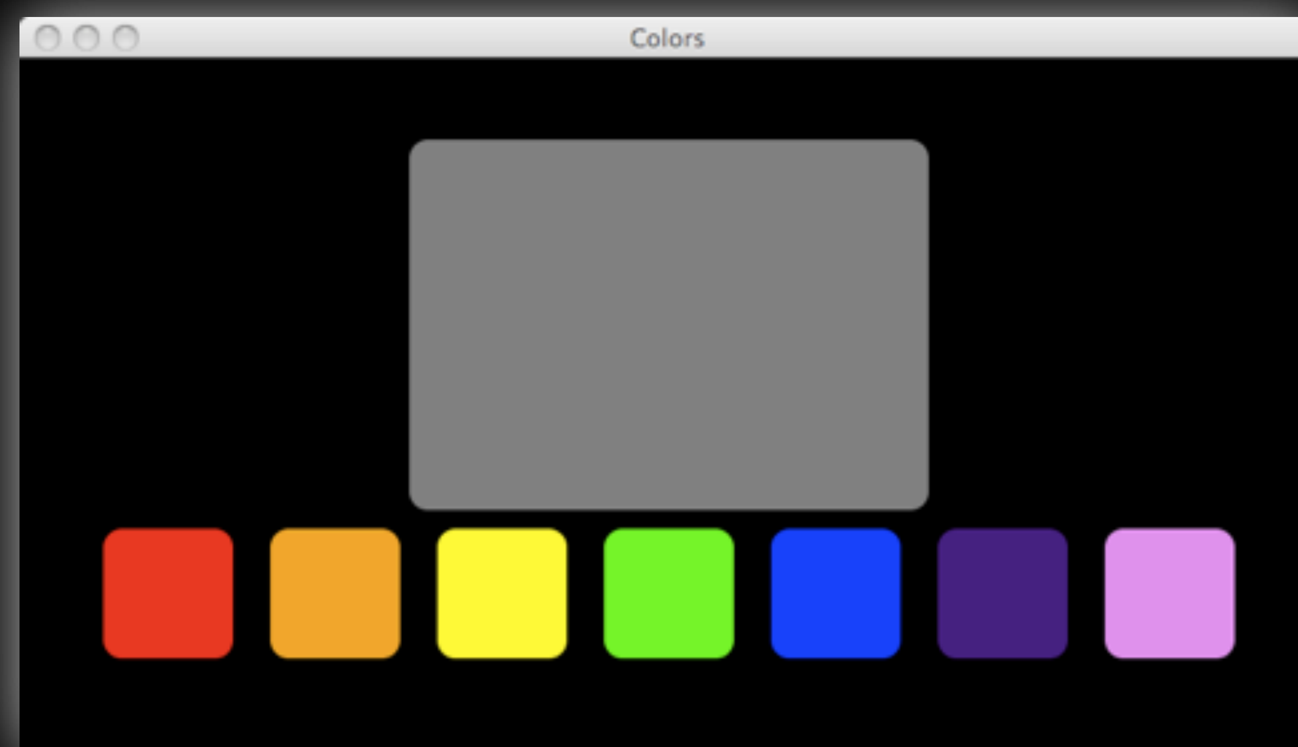
props.rb

```ruby
selection :id => "selection"
%w{ red orange yellow green blue indigo violet }.each do |color|
  sample :background_color => color
end
```

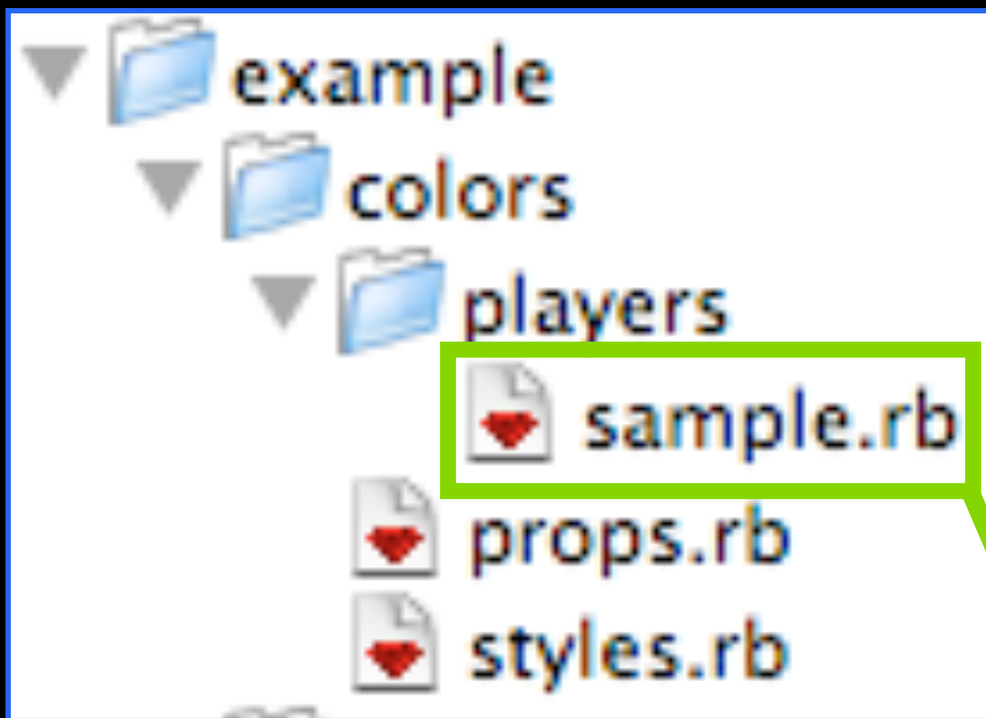
styles.rb

```ruby
selection {
  width "100%"
  left_margin "30%"
  right_margin "30%"
  height 200
  background_color :gray
  rounded_corner_radius 10
}

sample {
  width 90
  height 90
  rounded_corner_radius 10
  margin 10
}
```

on screen appearance



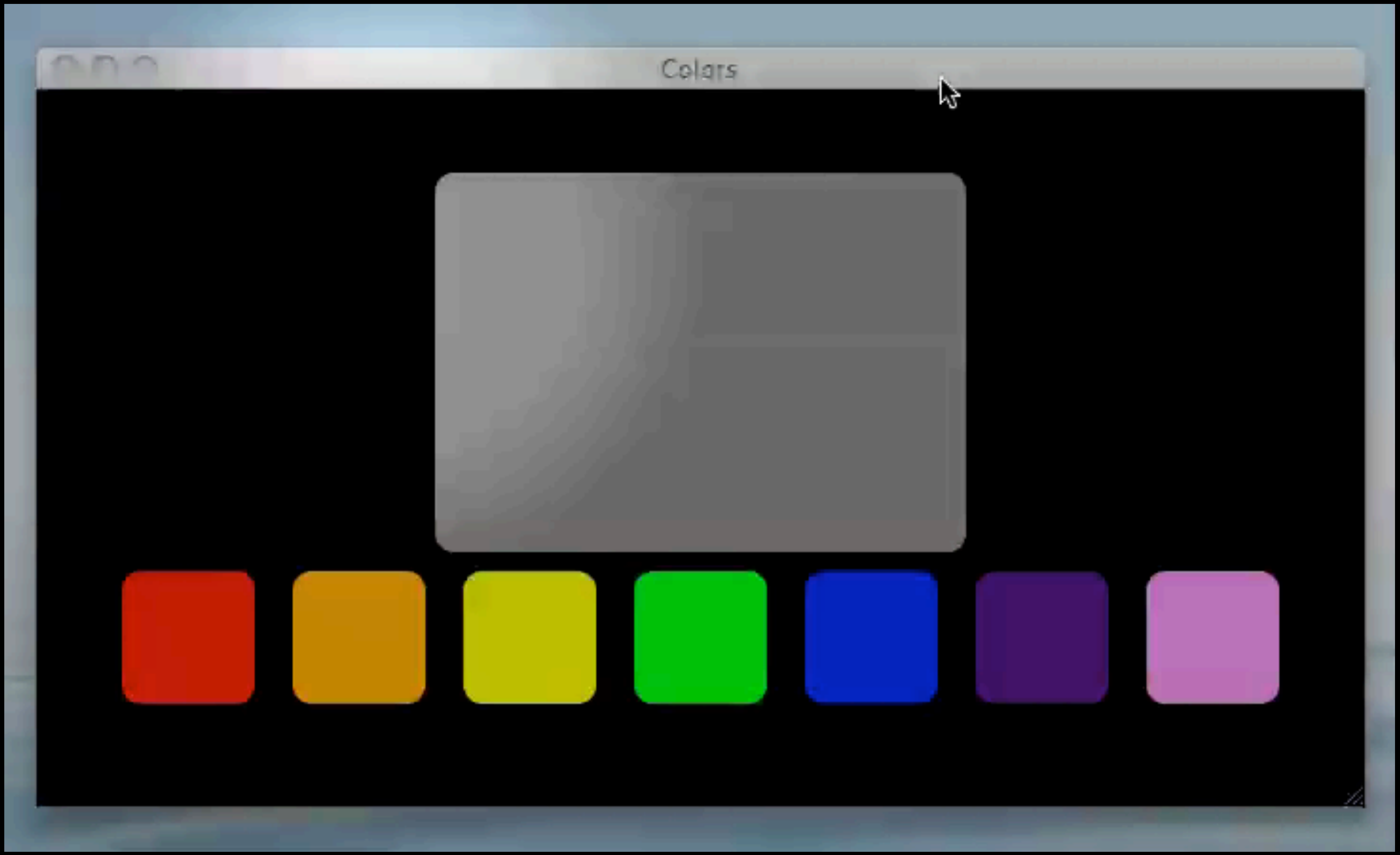
Adding Player



remember... the colored props are named "sample"

```ruby
module Sample
  def mouse_clicked(e)
    selection = scene.find("selection")
    color = self.style.background_color
    selection.style.background_color = color
  end
end
```

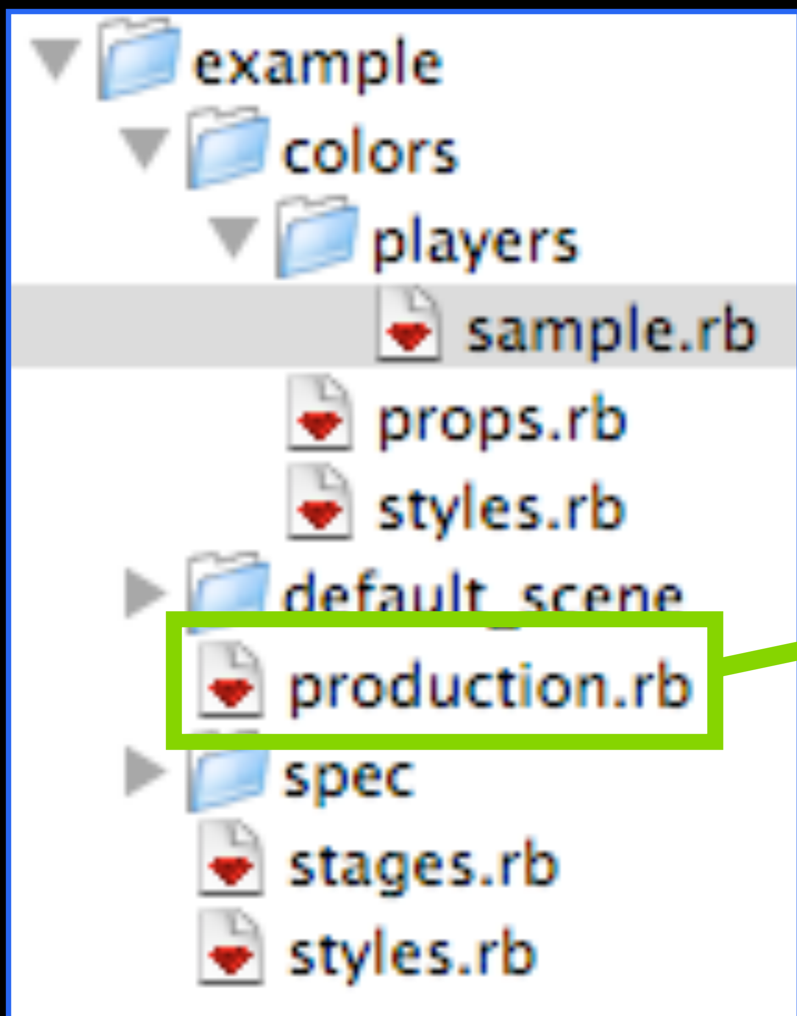
Colors in Action



Prop Hooks

- mouse_clicked
- mouse_entered
- mouse_exited
- mouse_pressed
- mouse_released
- mouse_dragged
- mouse_moved
- key_typed
- key_pressed
- key_released

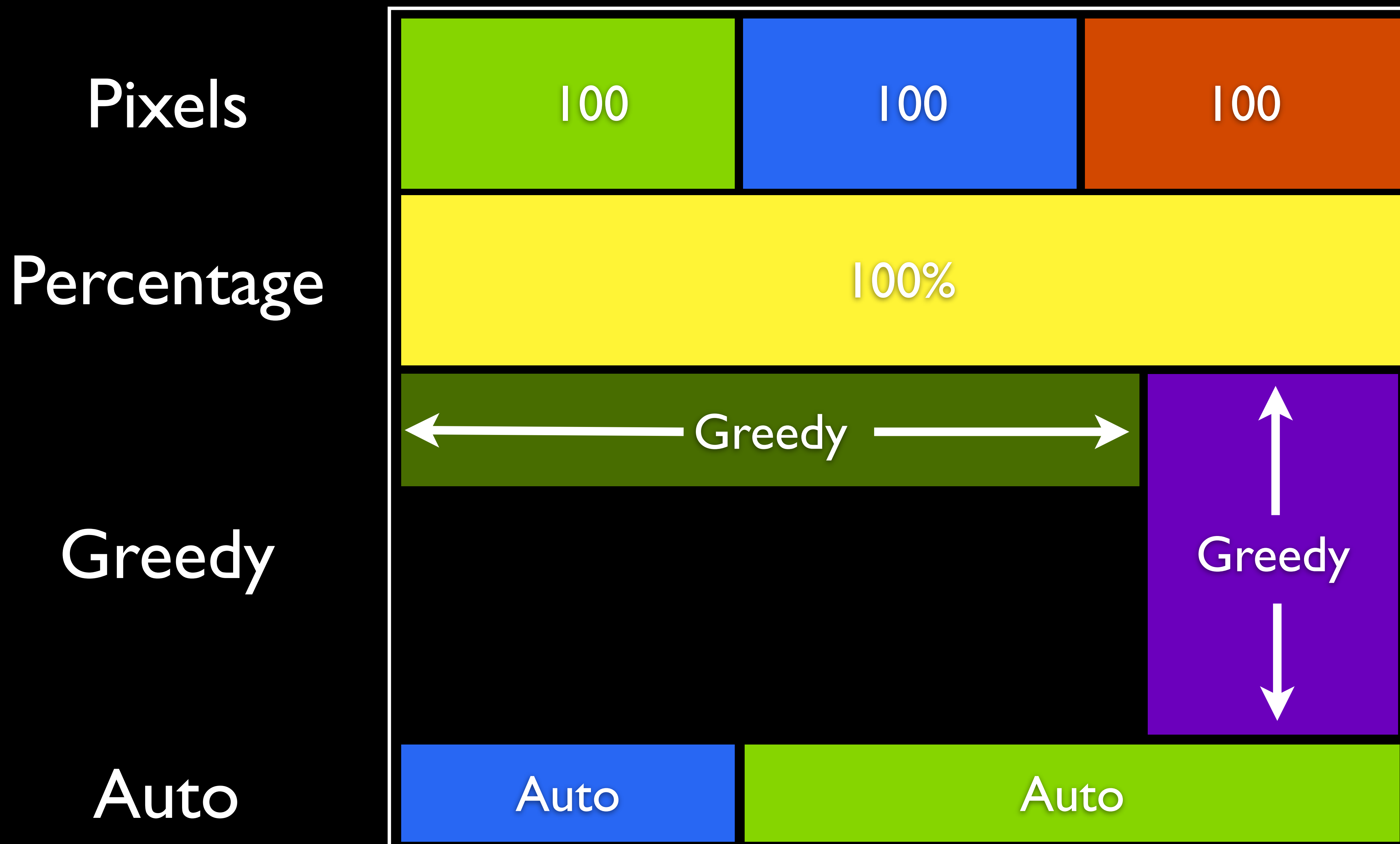
Production Hooks



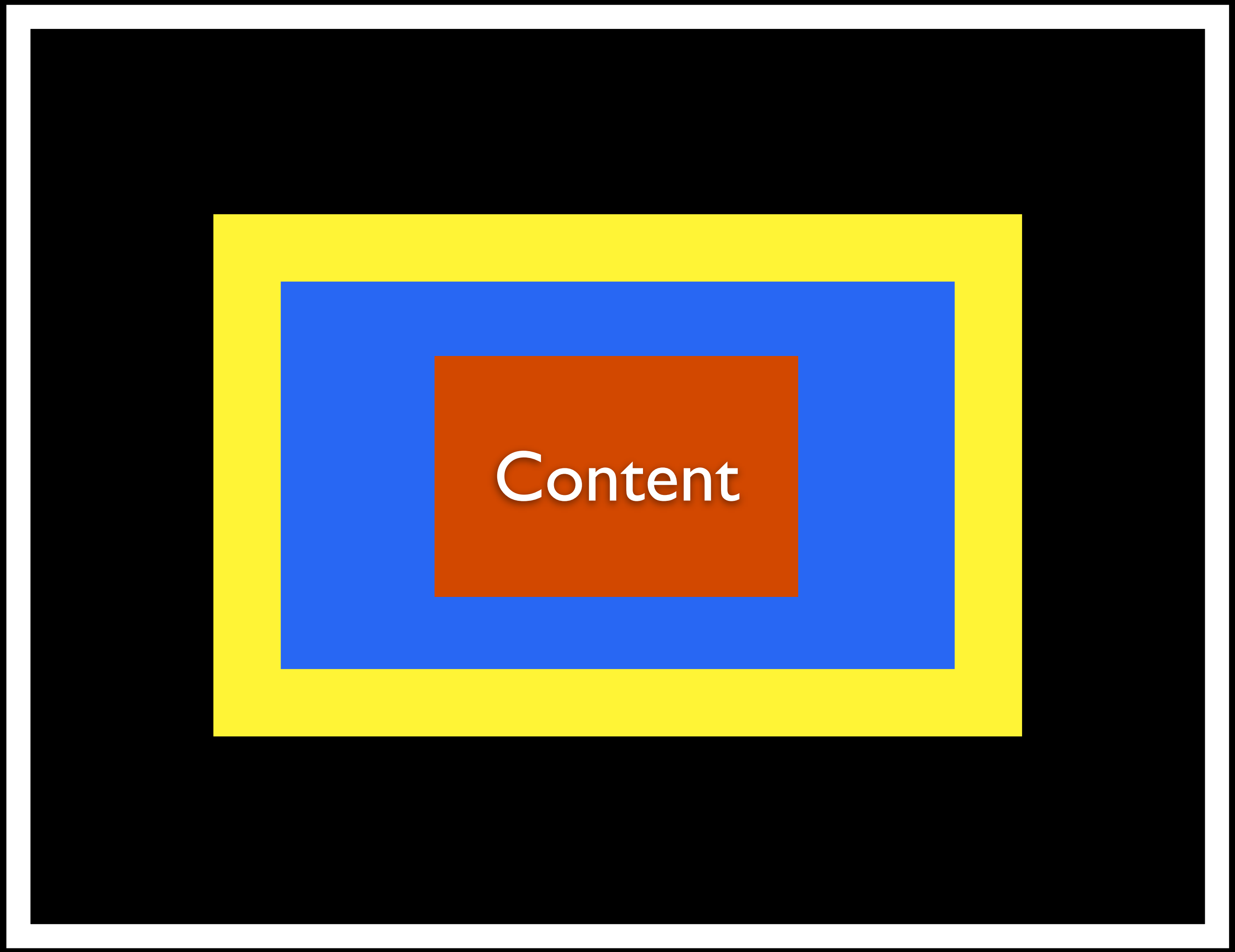

```ruby
module Production
#   def production_opening
#   end
#
#   def production_loaded
#   end
#
#   def production_opened
#   end
#
#   def allow_close?
#     return true
#   end
#
#   def production_closing
#   end
#
#   def production_closed
#   end
end
```

Limelight Styling

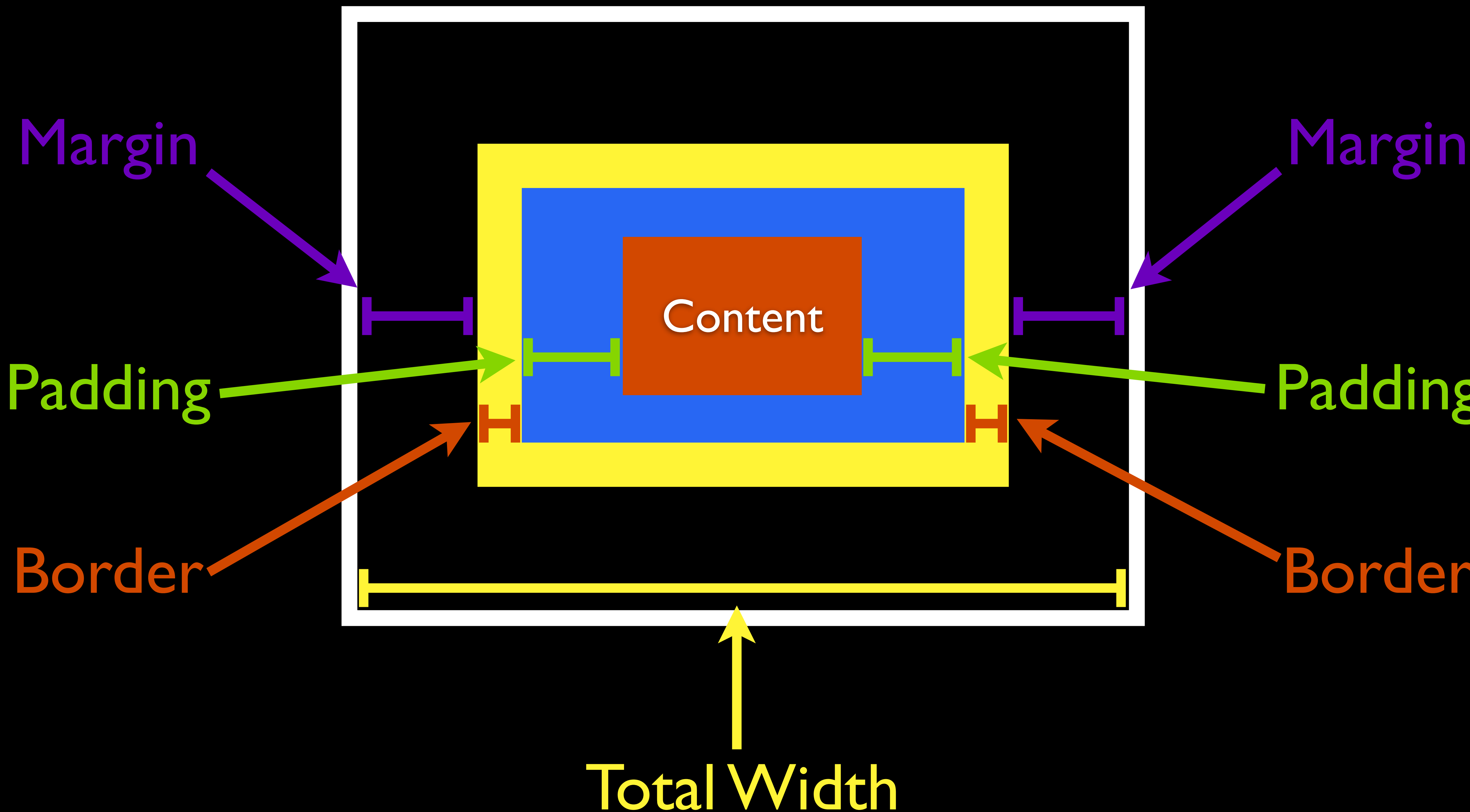
Sizing Props



Insets

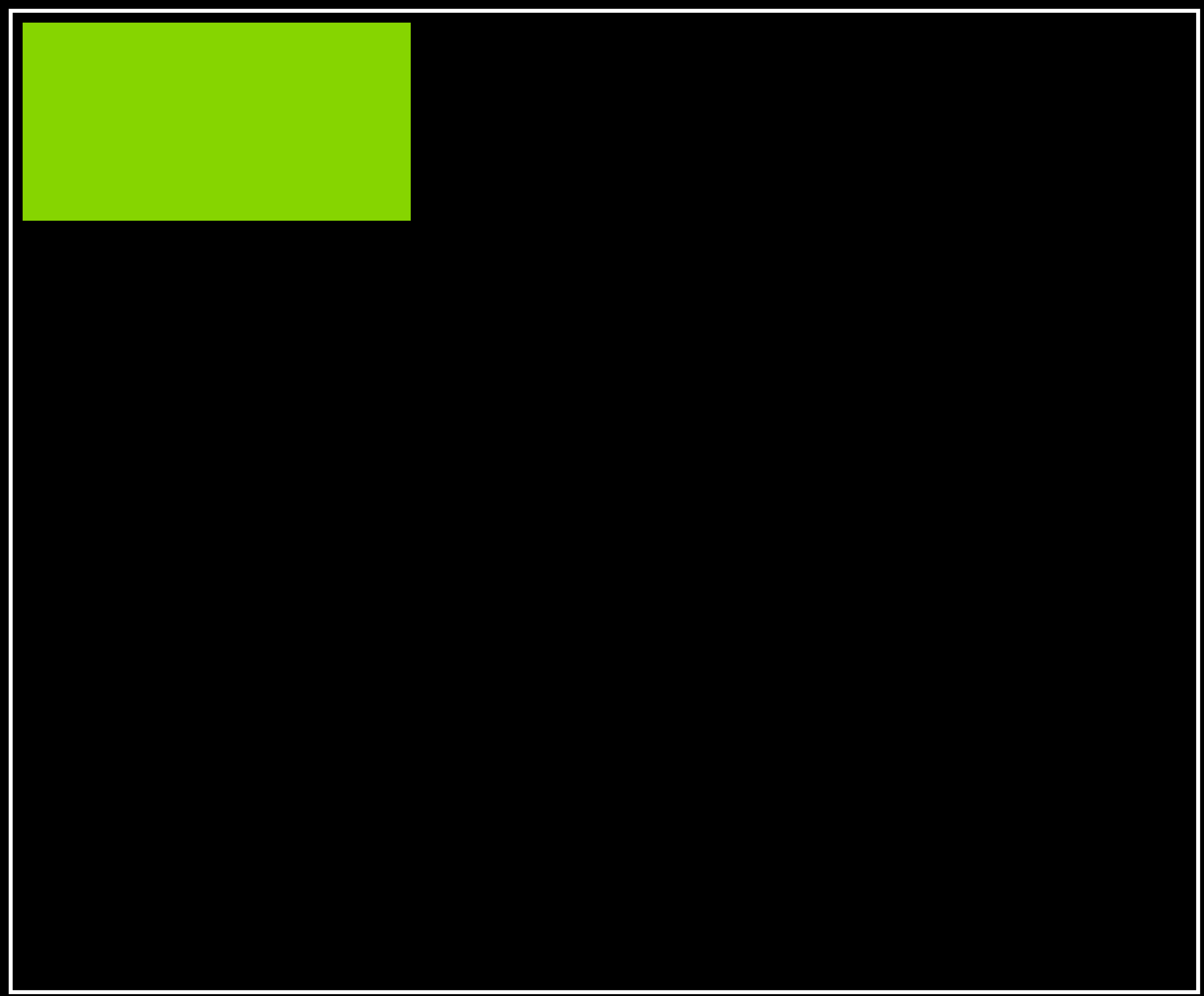


Insets

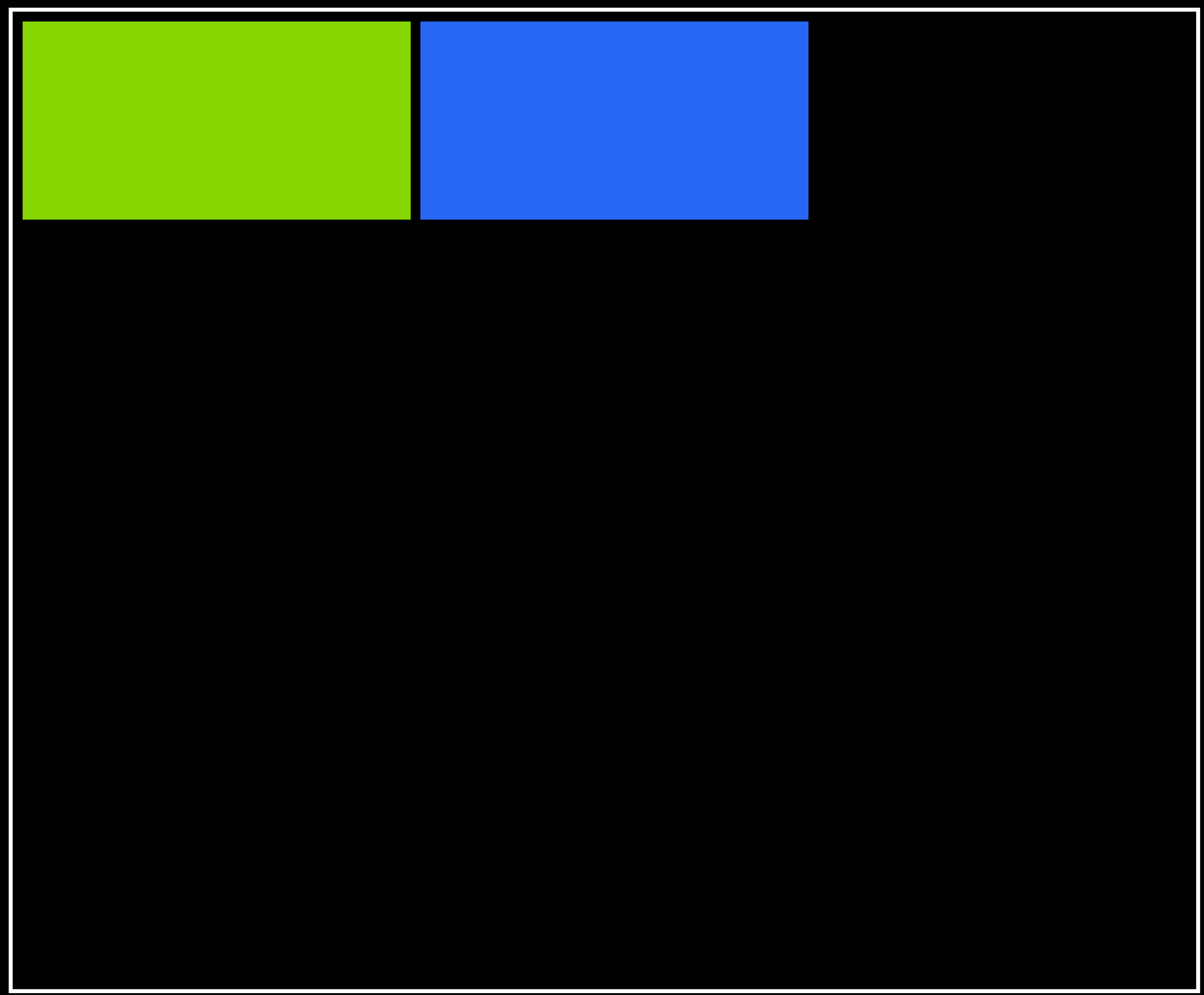

Margin
Margin
Content
Padding
Padding
Border
Border
Total Width

Simple Layout

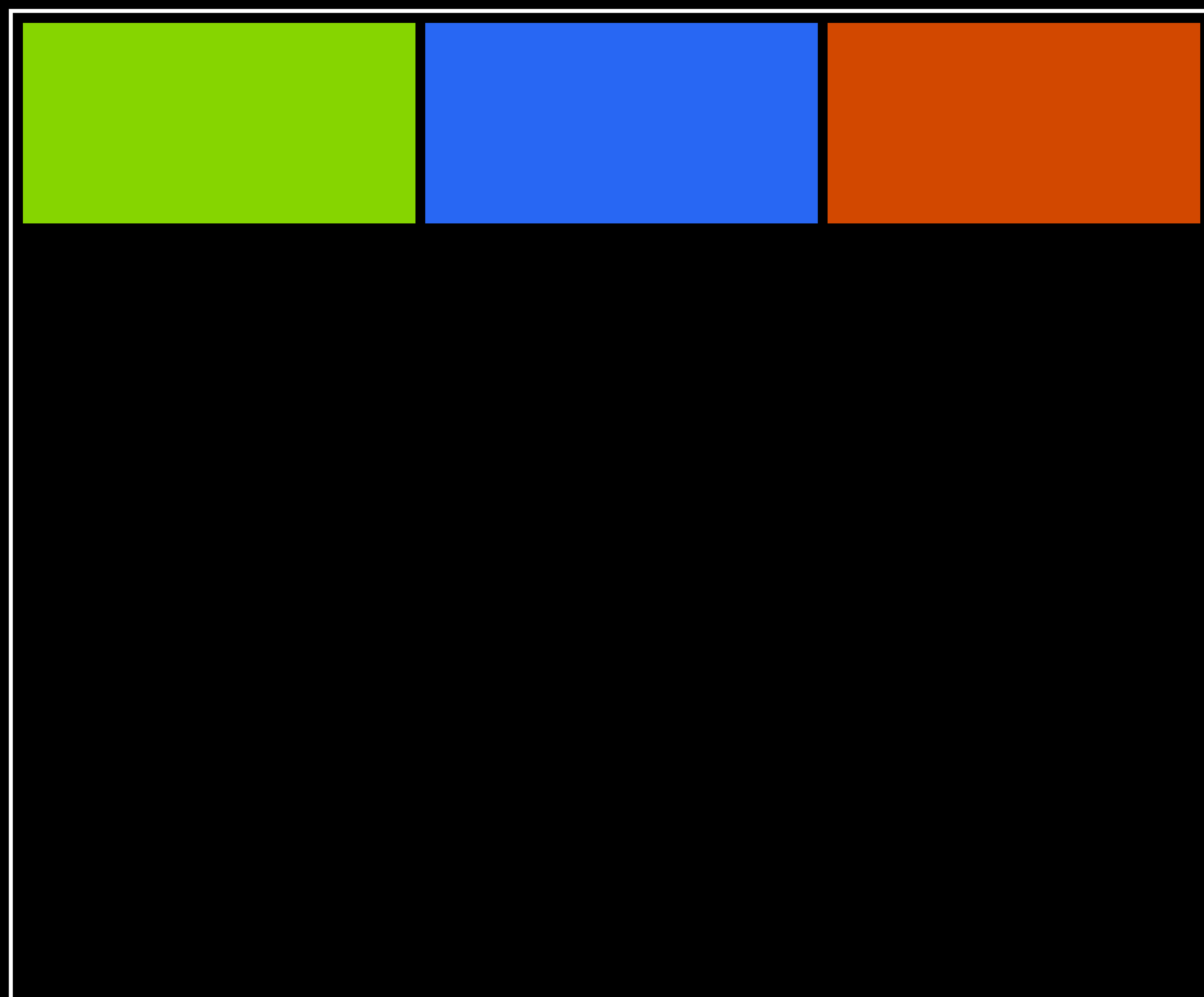
Simple Layout



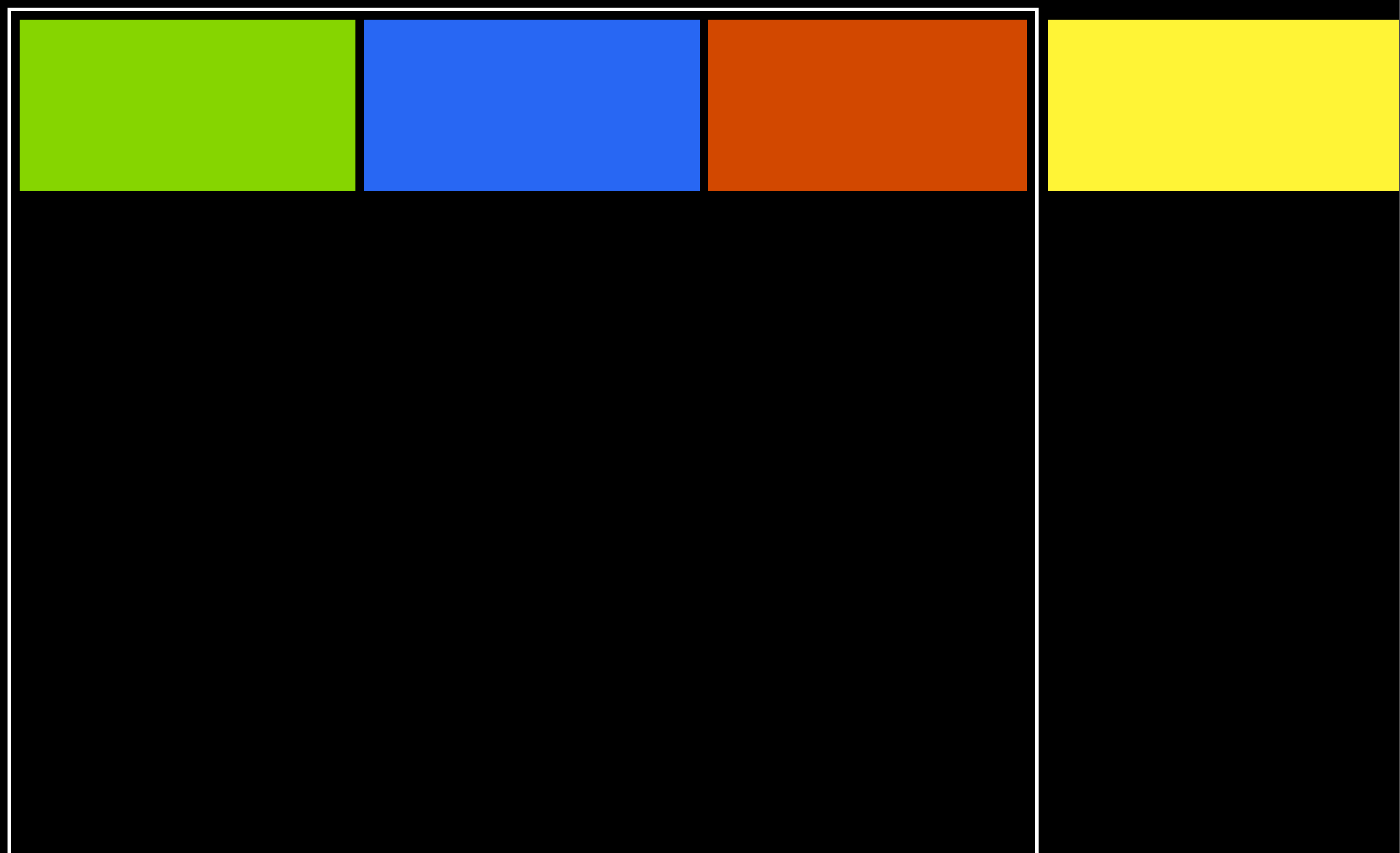
Simple Layout



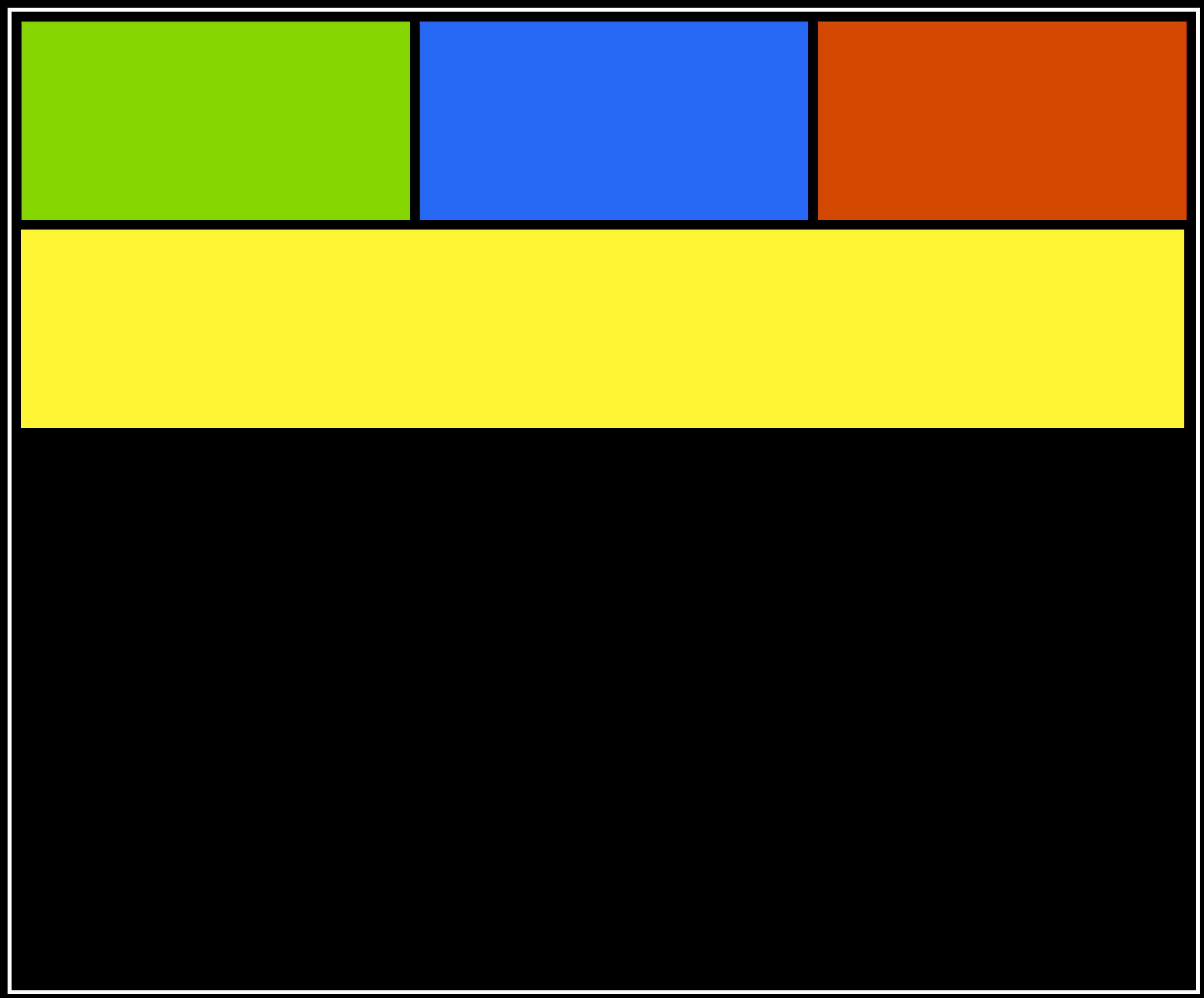
Simple Layout



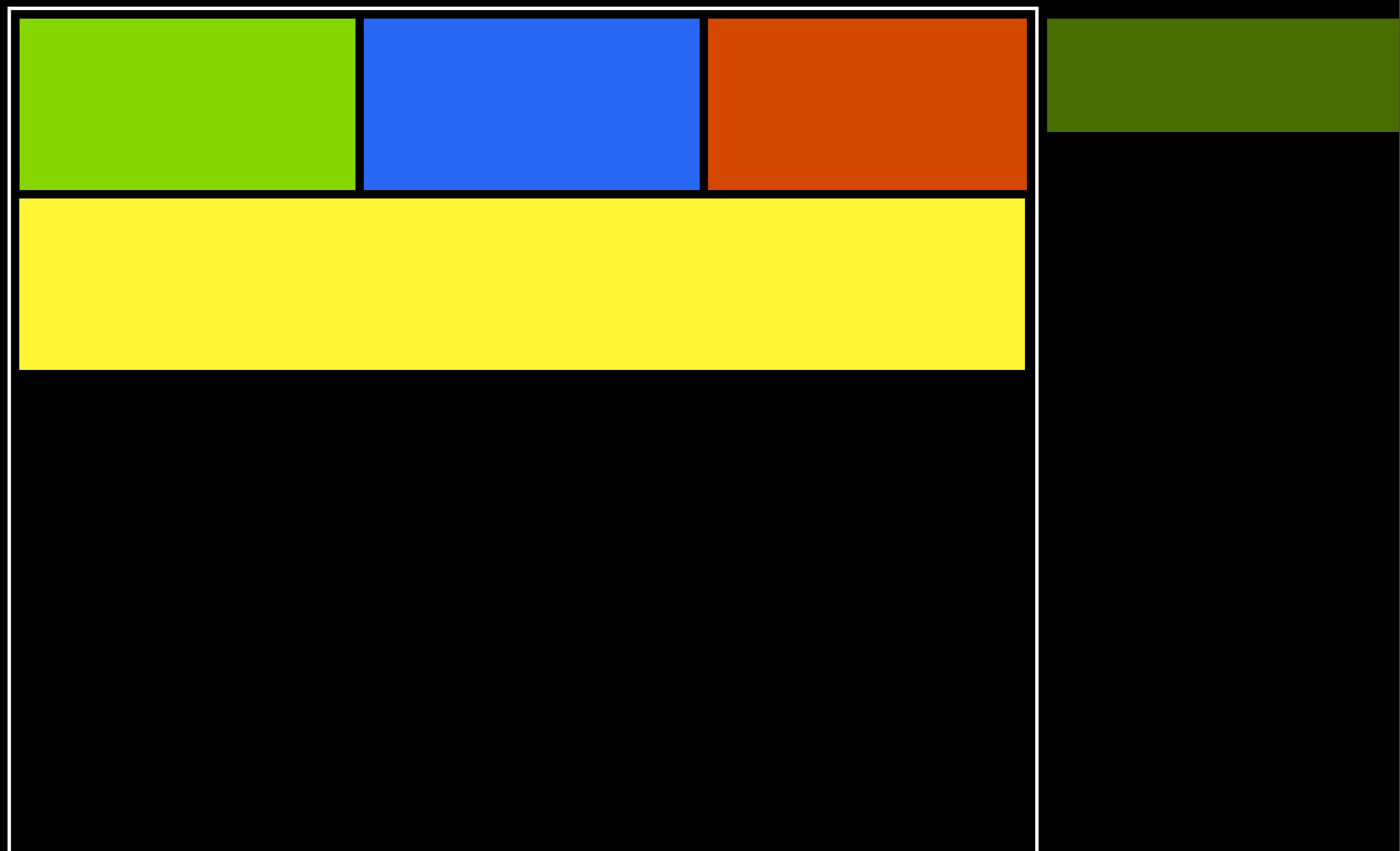
Simple Layout



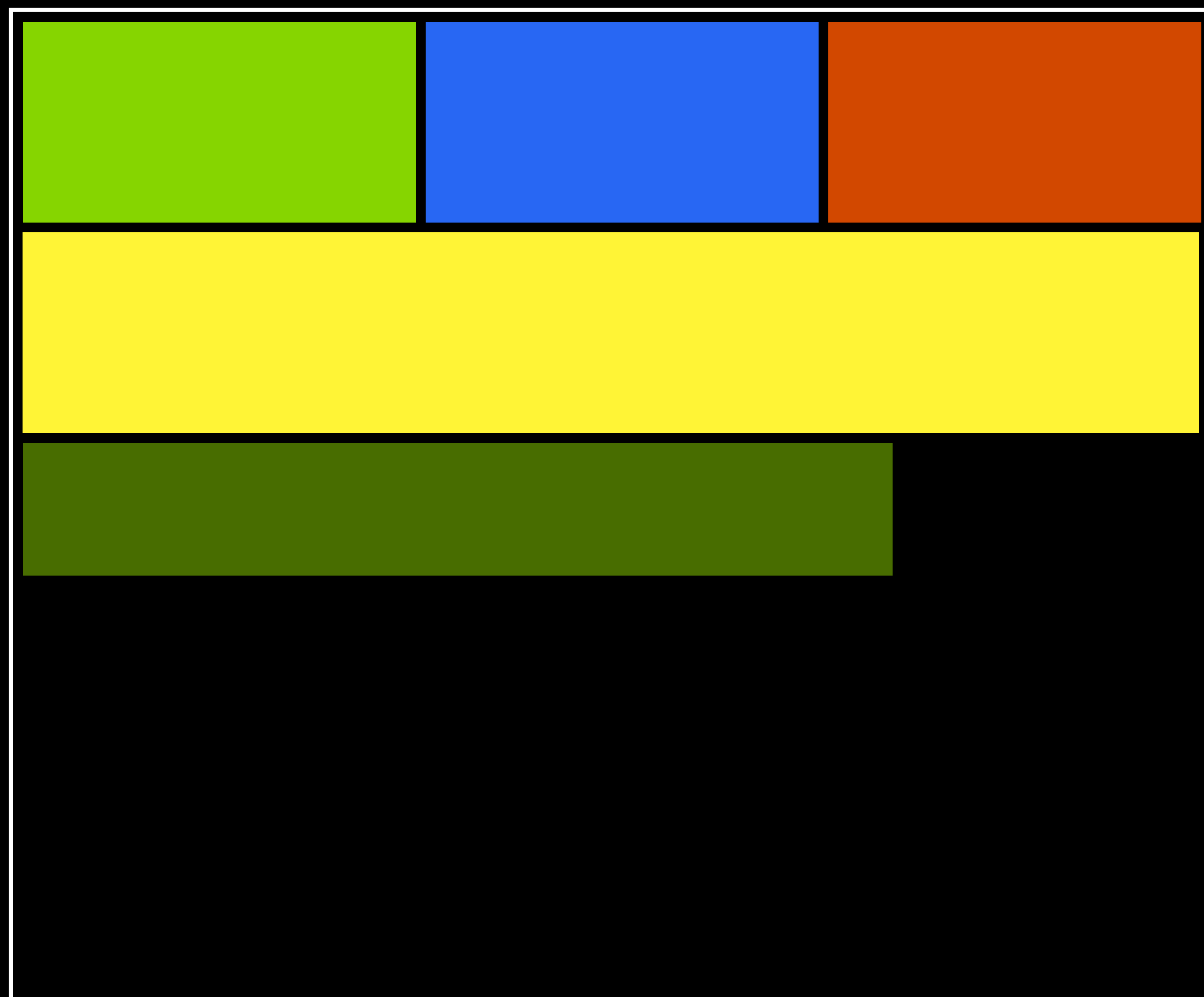
Simple Layout



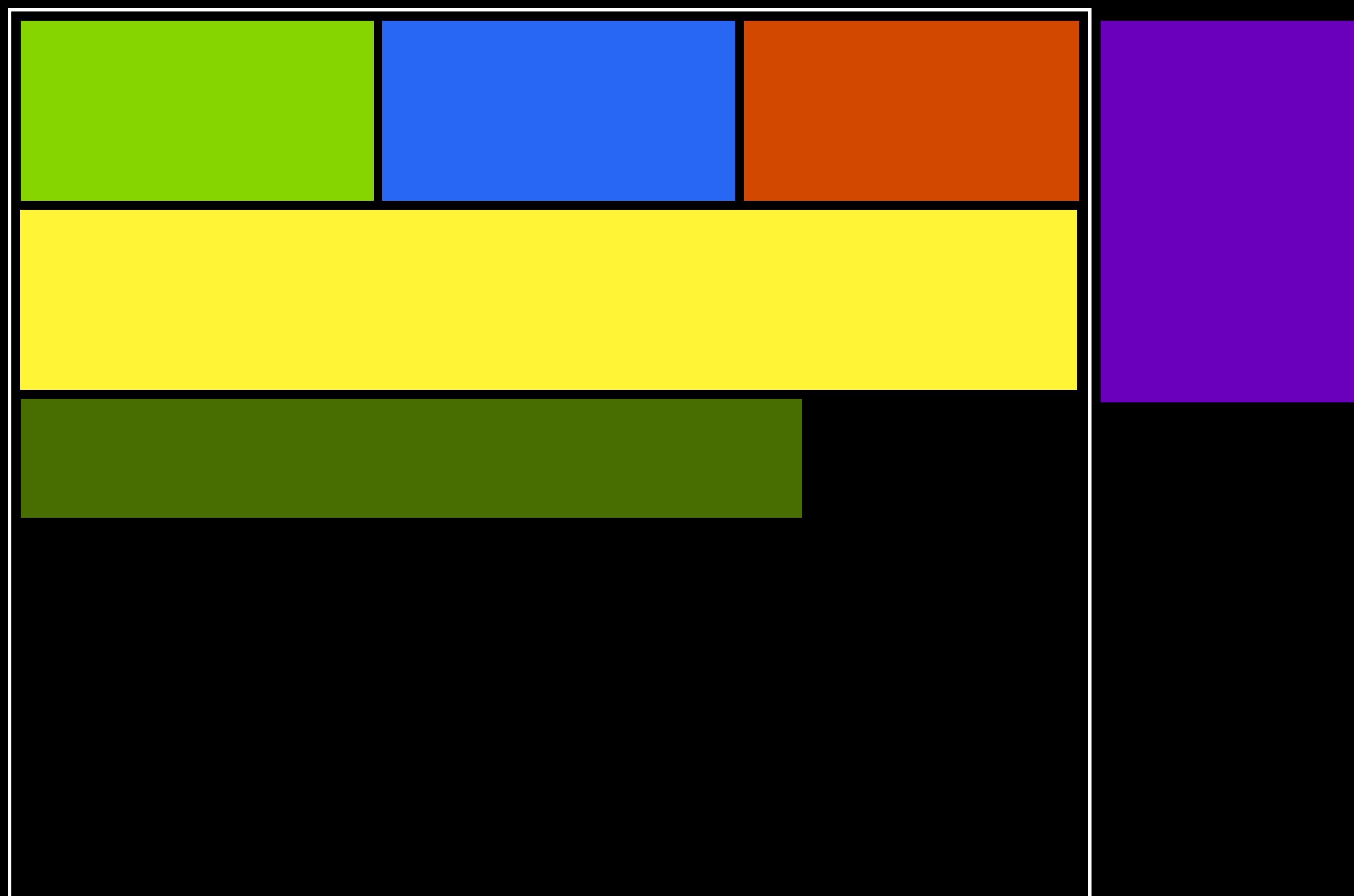
Simple Layout



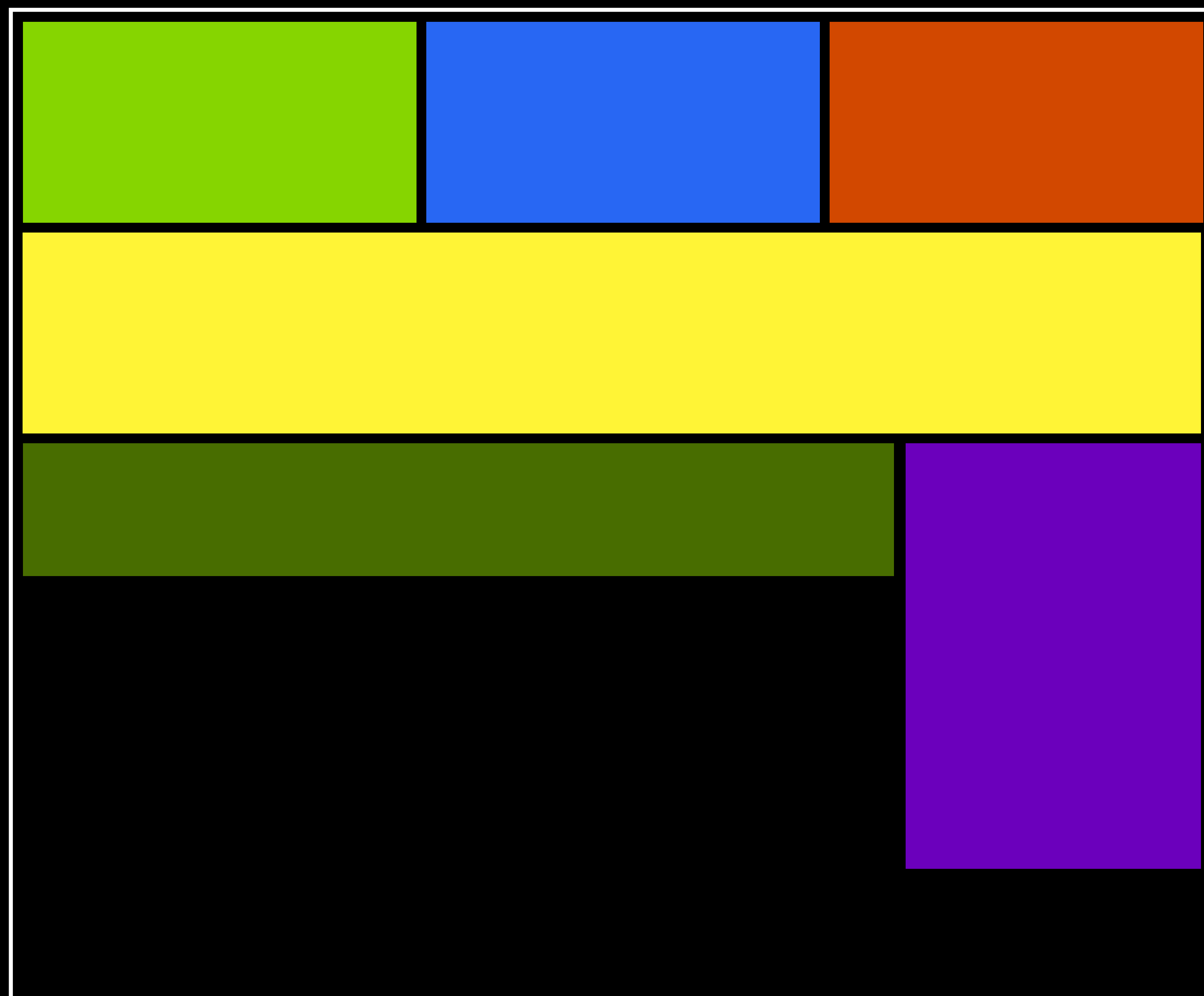
Simple Layout



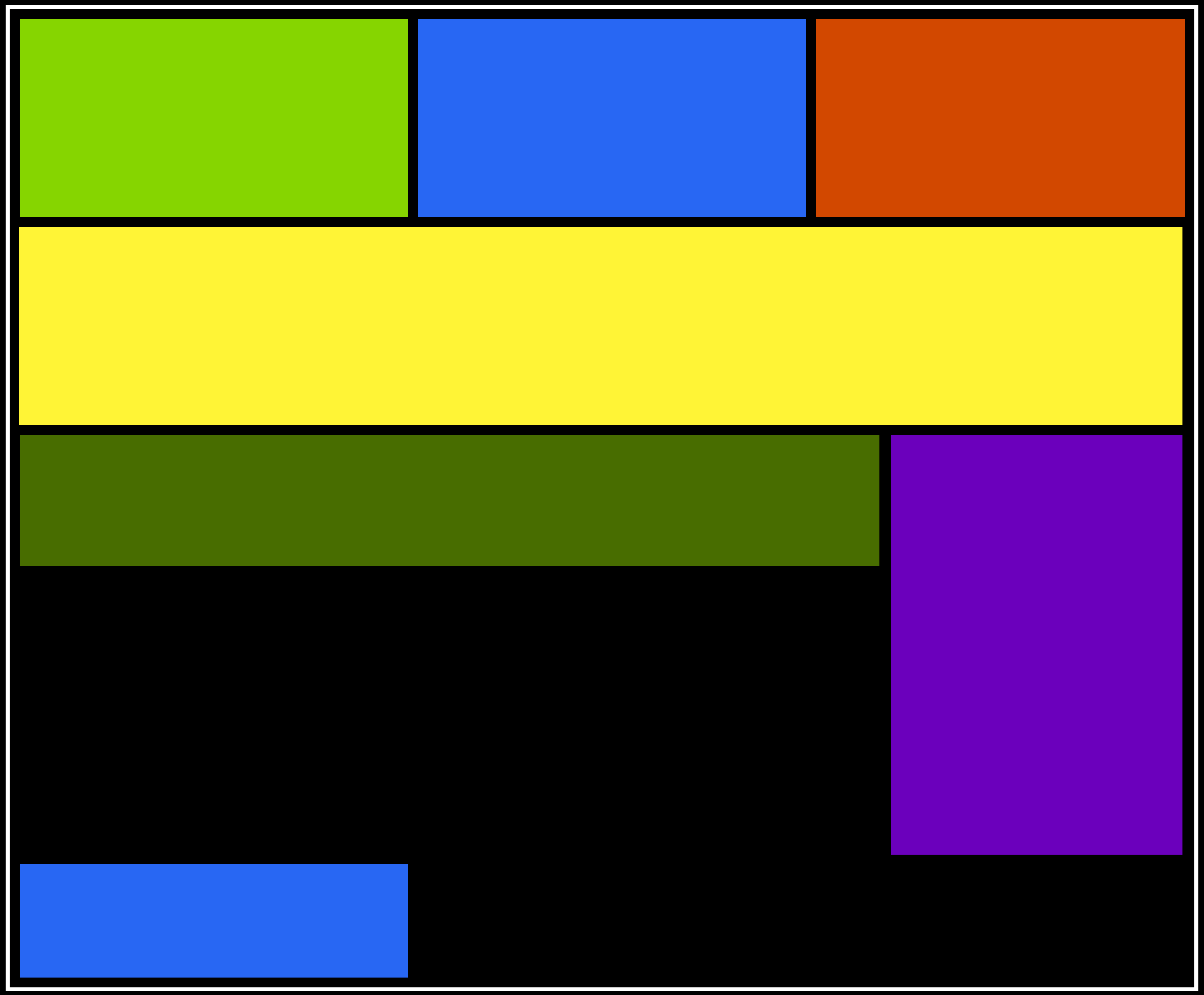
Simple Layout



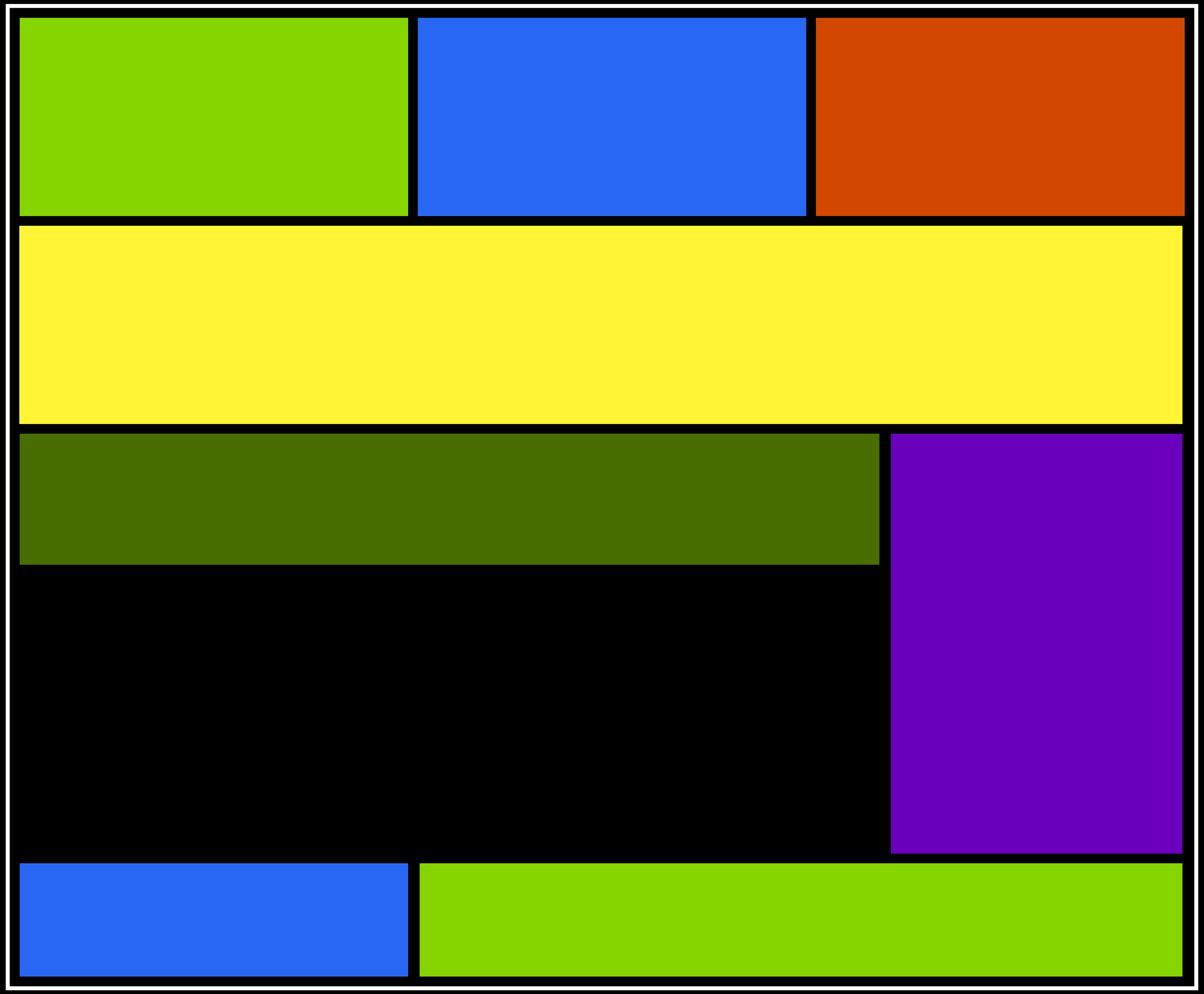
Simple Layout



Simple Layout



Simple Layout



Rounded Corners

Text

Font Face: All Fonts On Your SYSTEM Are Supported

Font Size: Make text as big (or small) as you want

Font Style: **Bold**, *Italic*, ***Bold Italic***

Text Color: Any color of the rainbow, and more

Backgrounds

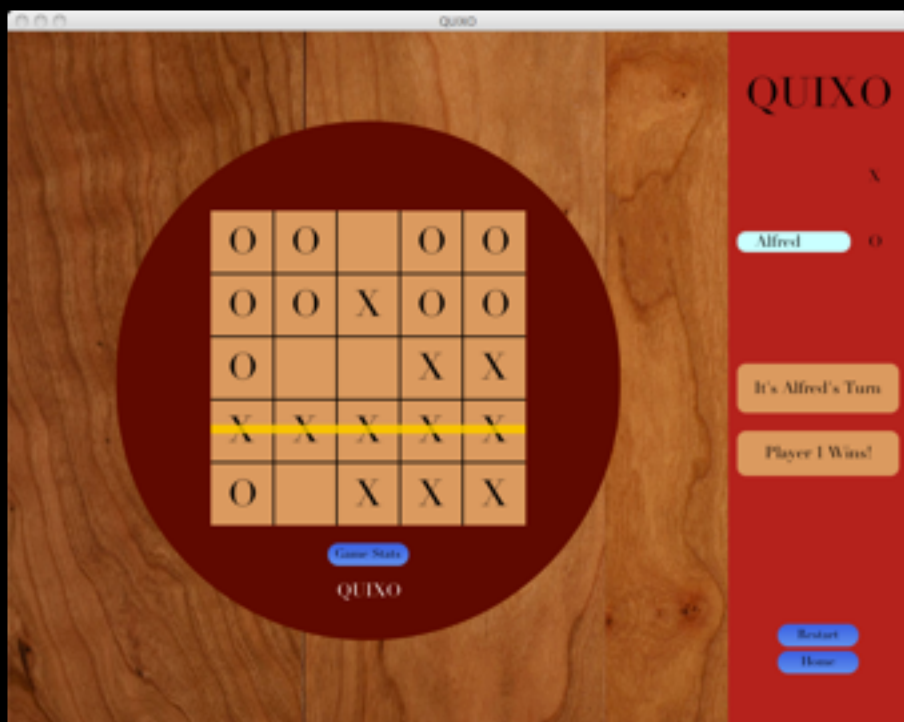
Background Image

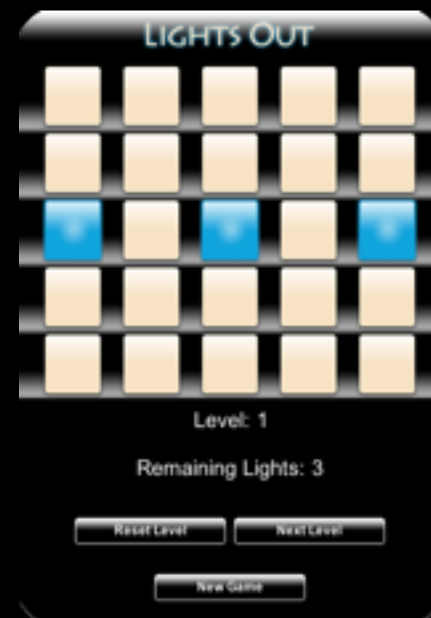
Solid Color Background

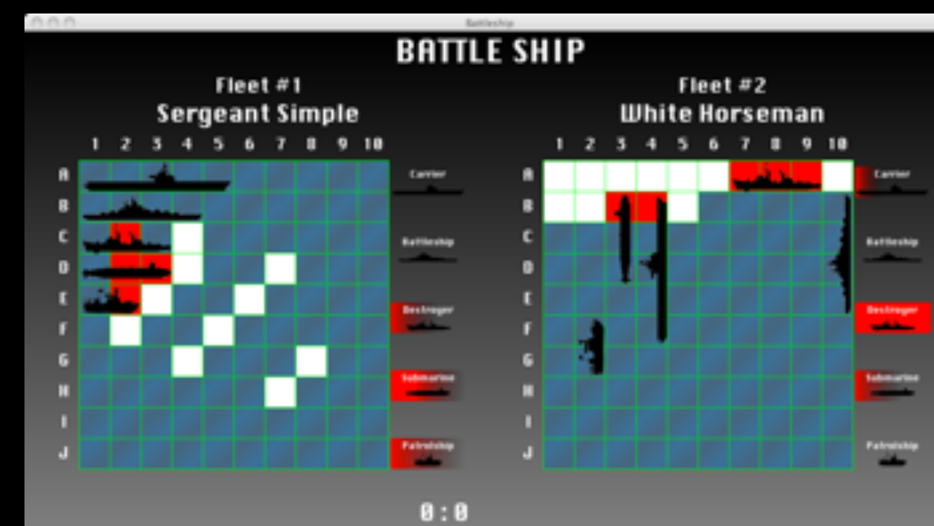
2 Color Gradient Background


Things You Can Build With Limelight
Copyright 2009 Micah Martin

Games?

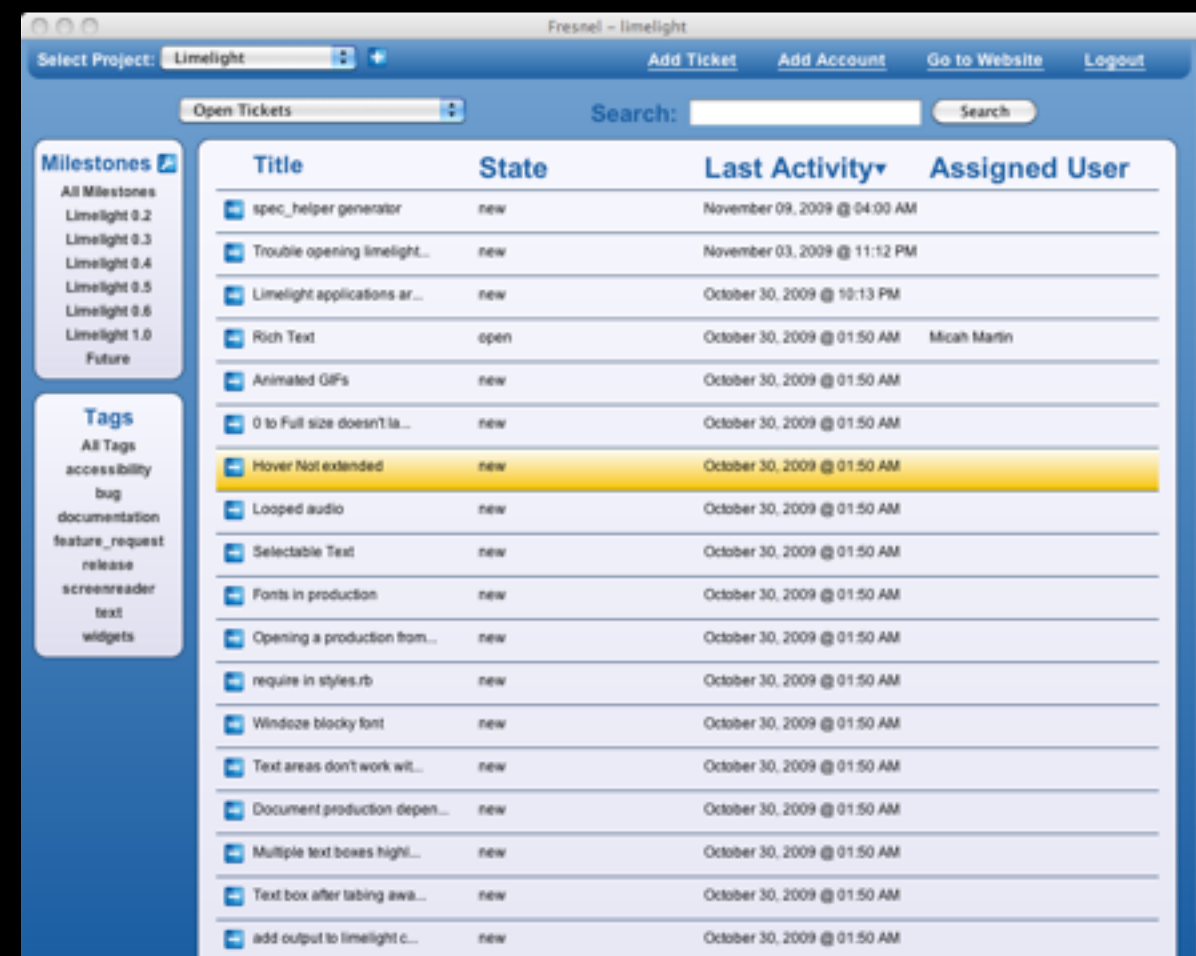

QUIXO
Alfred
It's Alfred's Turn
Player 1 Wins!
QUIXO
Restart
Home


LIGHTS OUT
Level: 1
Remaining Lights: 3
Reset Level
Next Level
New Game


BATTLE SHIP
Fleet #1
Sergeant Simple
Fleet #2
White Horseman
0 : 0


HANGMAN
New Game
Computer AIs
Close
P H A E T O N
Yes, there are some H's. The hangman is saved!
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
limelight

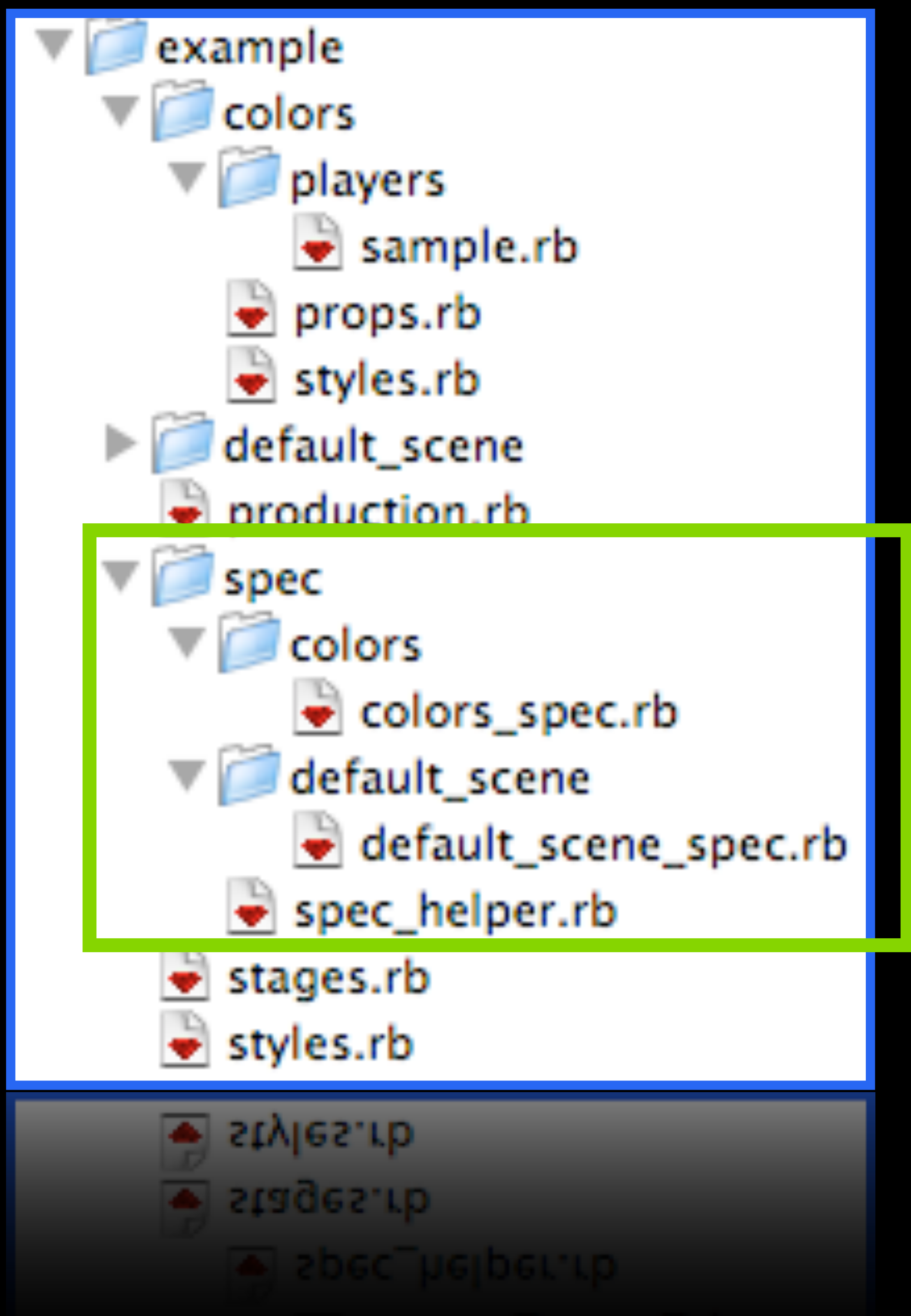
Business Apps



RIA (Rich Internet Applications)


Testing
Limelight
Productions
Copyright 2009 Micah Martin

RSpec: Unit Testing



- RSpec files generated for you
- 1 spec directory per scene by convention
- Limelight support for RSpec makes testing simple

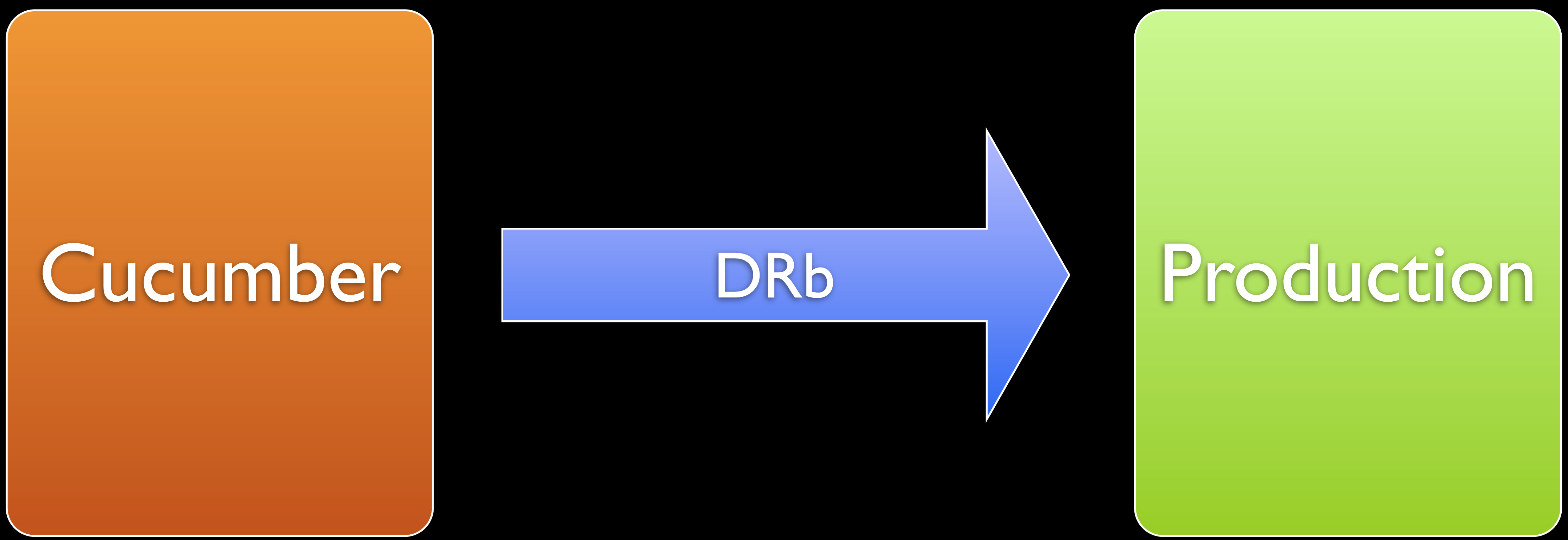
Colors Spec

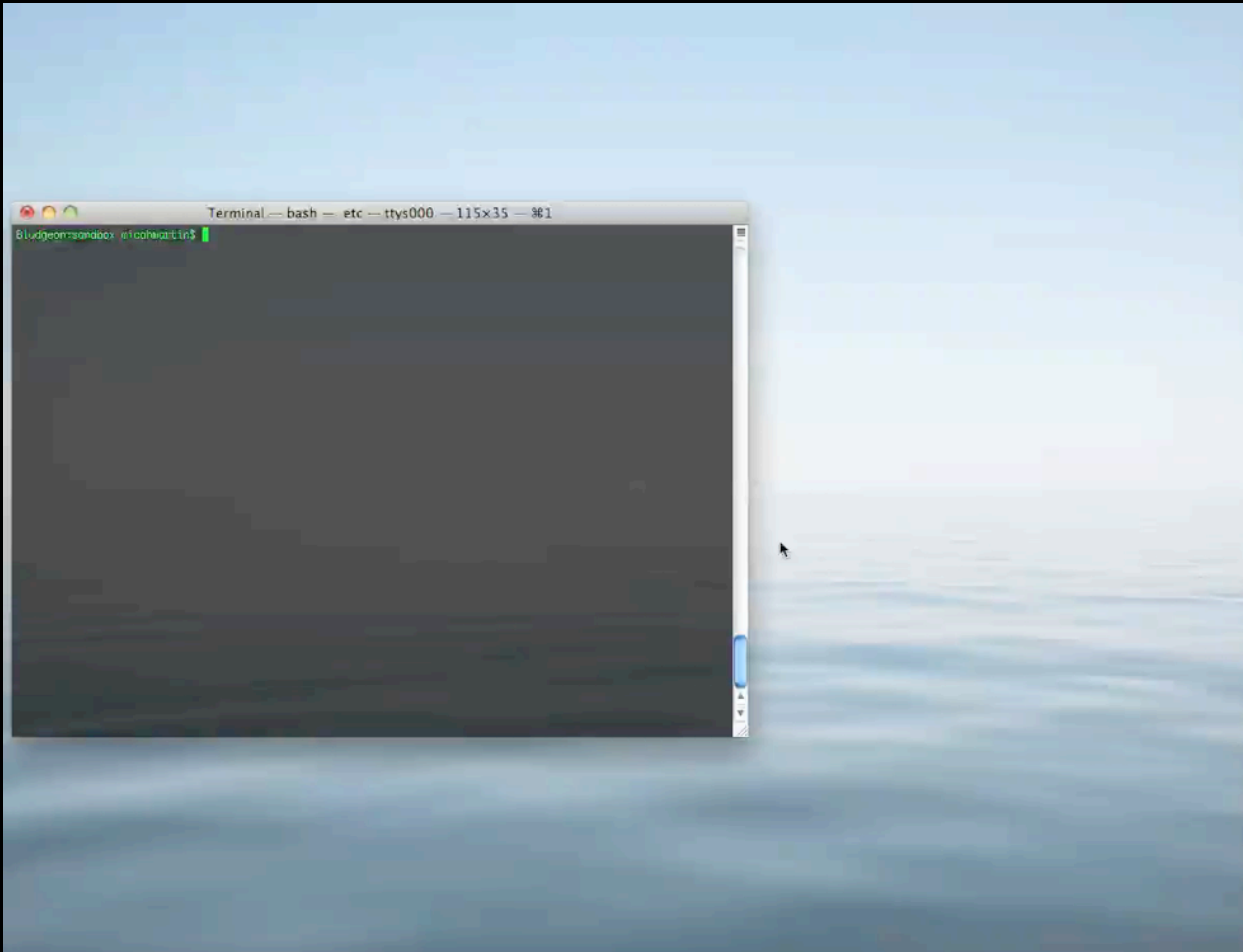
includes Limelight RSpec “goodness”

```
require File.expand_path(File.dirname(__FILE__) + "/../spec_helper")

describe "Colors Scene" do

  uses_limelight :scene => "colors"

  it "should change selection color when I click on red" do
    red_sample = scene.find_by_name("sample")[0]
    red_sample.mouse_clicked(nil)
    selection = scene.find("selection")
    selection.style.background_color.should == "#ff0000ff"
  end

end
```

provided helper method

Cucumber: Acceptance Testing

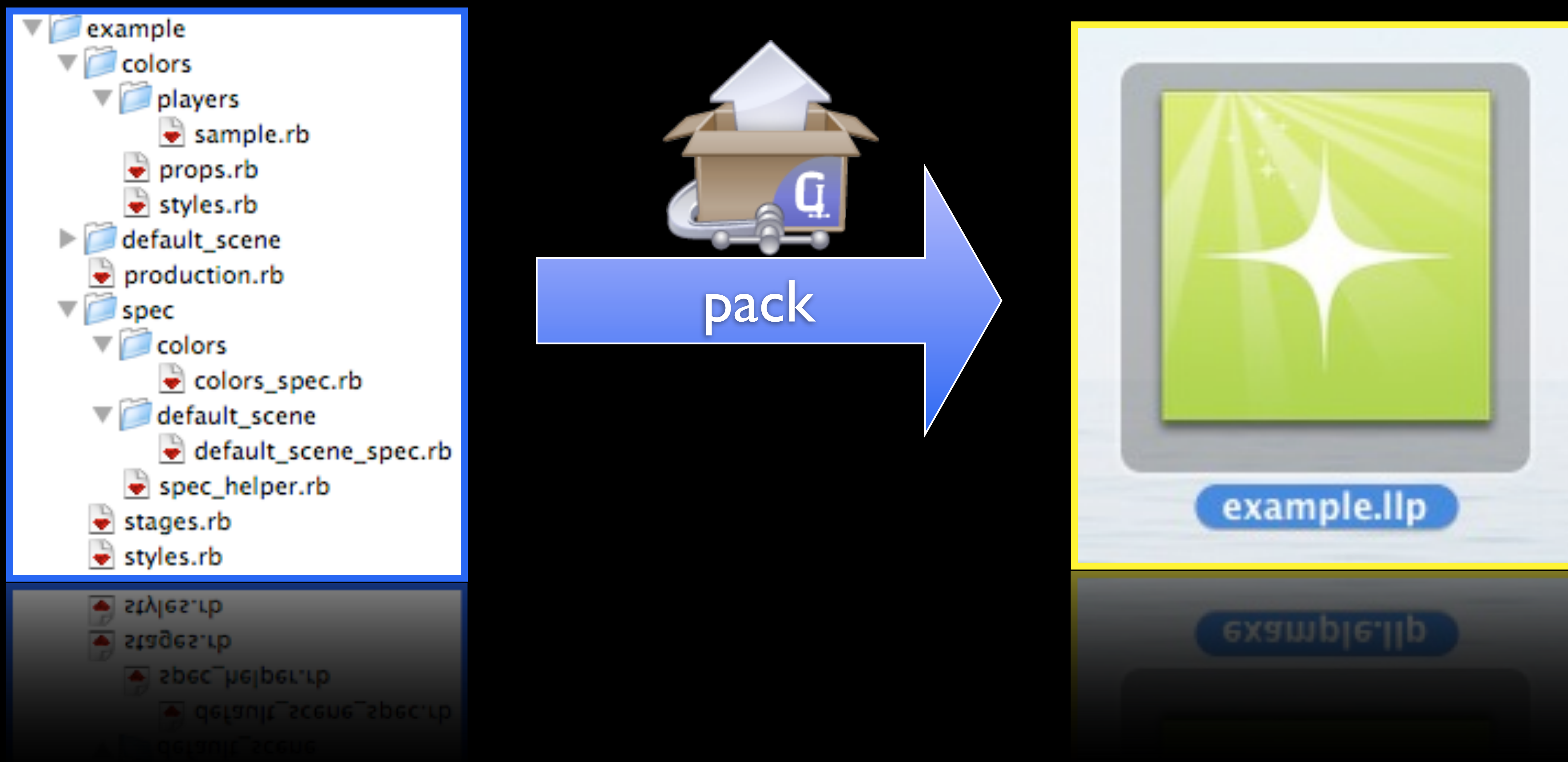



Terminal — bash — etc — ttys000 — 115×35 — ⌘1
Bludgeon:sandbox micahmartin$

Packaging and Publishing Productions

Packaging

```
$ jruby -S limelight pack example
```

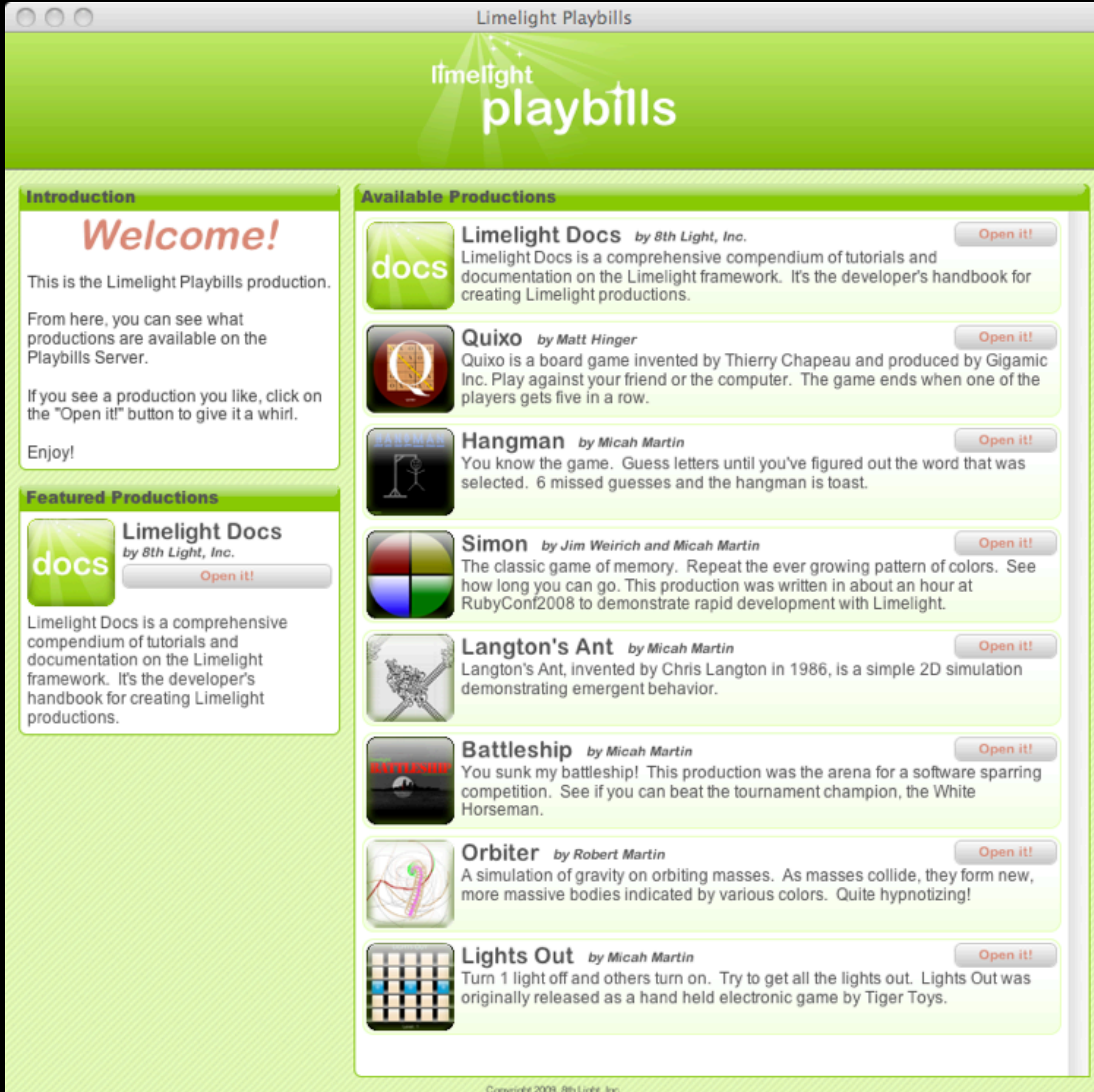


Publishing

The Web

example.lll

```
http://mysite.com/example.llp
```


Limelight Playbills
limelight playbills
Introduction
Welcome!
This is the Limelight Playbills production.
From here, you can see what productions are available on the Playbills Server.
If you see a production you like, click on the "Open it!" button to give it a whirl.
Enjoy!
Featured Productions
docs
Limelight Docs
by 8th Light, Inc.
Open it!
Limelight Docs is a comprehensive compendium of tutorials and documentation on the Limelight framework. It's the developer's handbook for creating Limelight productions.
Available Productions
docs
Limelight Docs by 8th Light, Inc.
Open it!
Limelight Docs is a comprehensive compendium of tutorials and documentation on the Limelight framework. It's the developer's handbook for creating Limelight productions.
Quixo by Matt Hinger
Open it!
Quixo is a board game invented by Thierry Chapeau and produced by Gigamic Inc. Play against your friend or the computer. The game ends when one of the players gets five in a row.
Hangman by Micah Martin
Open it!
You know the game. Guess letters until you've figured out the word that was selected. 6 missed guesses and the hangman is toast.
Simon by Jim Weirich and Micah Martin
Open it!
The classic game of memory. Repeat the ever growing pattern of colors. See how long you can go. This production was written in about an hour at RubyConf2008 to demonstrate rapid development with Limelight.
Langton's Ant by Micah Martin
Open it!
Langton's Ant, invented by Chris Langton in 1986, is a simple 2D simulation demonstrating emergent behavior.
Battleship by Micah Martin
Open it!
BATTLESHIP
You sunk my battleship! This production was the arena for a software sparring competition. See if you can beat the tournament champion, the White Horseman.
Orbiter by Robert Martin
Open it!
A simulation of gravity on orbiting masses. As masses collide, they form new, more massive bodies indicated by various colors. Quite hypnotizing!
Lights Out by Micah Martin
Open it!
Turn 1 light off and others turn on. Try to get all the lights out. Lights Out was originally released as a hand held electronic game by Tiger Toys.
Copyright 2009, 8th Light, Inc.

Installing Limelight

Download Installer

Run Installer

Limelight Publishing

- One Install (Limelight)
- Easy Updates
- Efficient/Minimal Downloads


Finale
Copyright 2009 Micah Martin

Limelight ...

is Ruby on the Desktop

is designed around a theater metaphor

uses a simple file-based structure

chock full of expressive DSLs

makes testing a breeze

productions are highly publishable

That's a Wrap!

limelight

http://limelight.8thlight.com

Micah Martin
@slagyr
8th Light, Inc.

8th light

http://8thlight.com

That’s a Wrap!

limelight

http://limelight.8thlight.com

Micah Martin
@slagyr
8th Light, Inc.

8th light

http://8thlight.com

HIRING
CRAFTSMEN