Wii! Playing with Ruby

James Britt

http://www.google.com/search?q=James+Britt

Play

Play

```
# Add these in to Wii-enable this c
require 'wii_api_manager'

require 'wiimotable'
require 'wiimote_event_listener'

require 'midi'

class DisplayController < Applicati
  set_model 'DisplayModel'
  set_view 'DisplayView'
  set_close_action :exit

  # mixin the module ...
  include Neurogami::Wiimotable
```

Play

Hardware

Wii CPU

Wii IR sensor bar

Wii remote (“Wiimote”)

Hardware

Wii CPU

Wii IR sensor bar

Wii remote (“Wiimote”)

Hardware

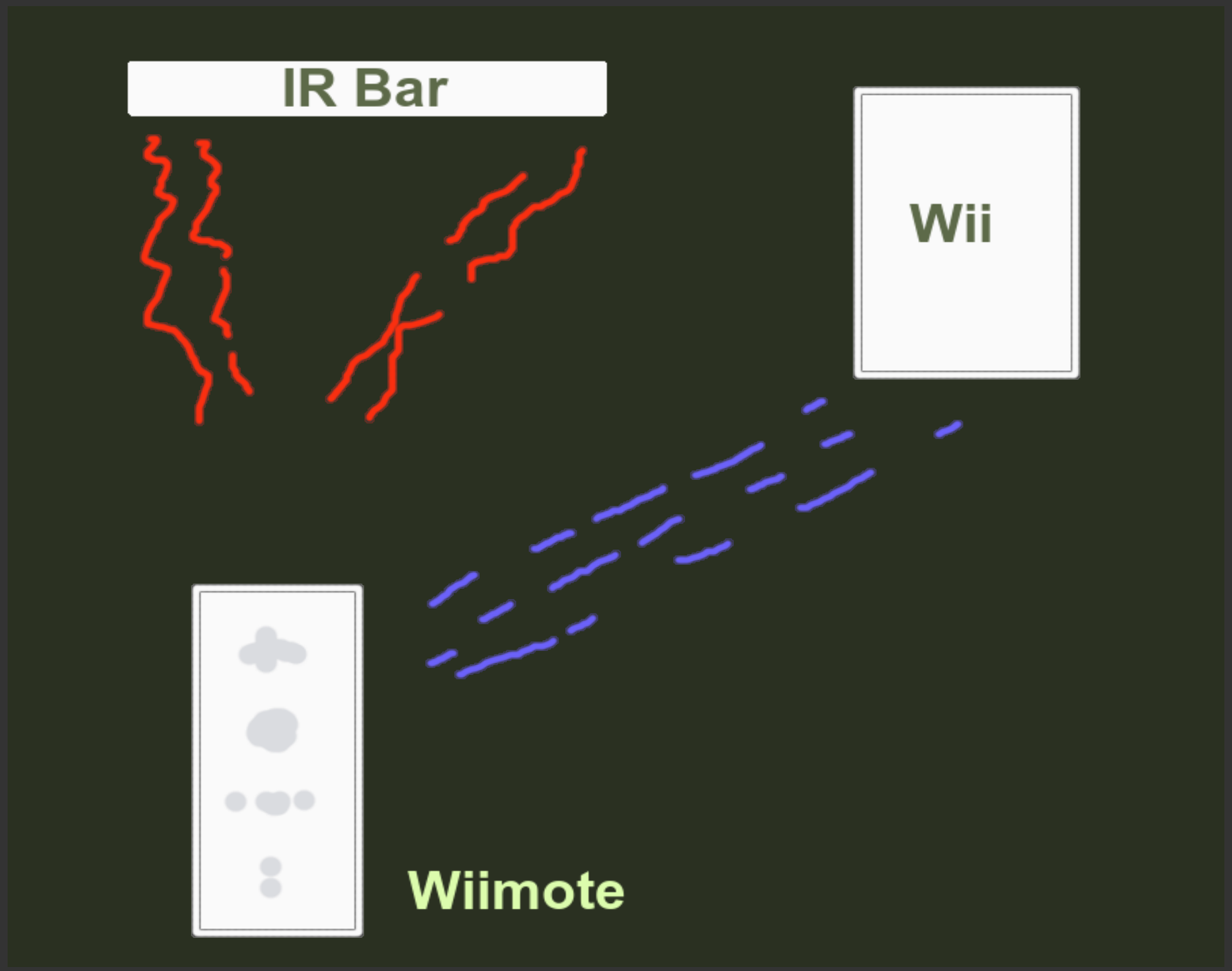
Wii CPU

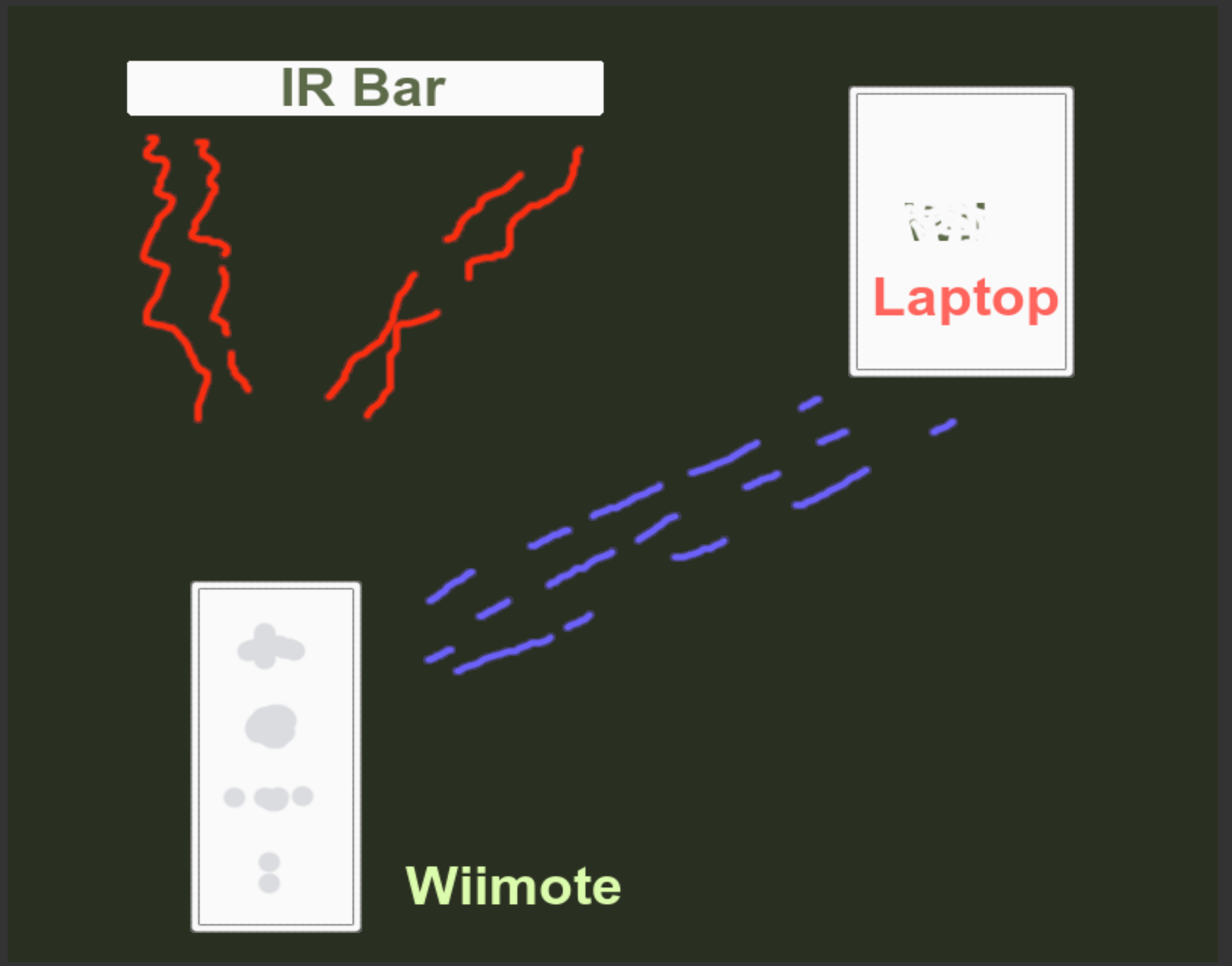
Wii IR sensor bar

Wii remote (“Wiimote”)

Nunchuck

Balance board


IR Bar
Wii
Wiimote


IR Bar
Laptop
Wiimote

Getting yours

Buy a Nintendo Wiimote: USD 40.00

Buy a 3rd-party sensor bar: USD 20.00
(PSE112 from Psyclone is nice!)

Build your own sensor bar
(Google + solder iron are your friends.)

Making the connection

Making the connection

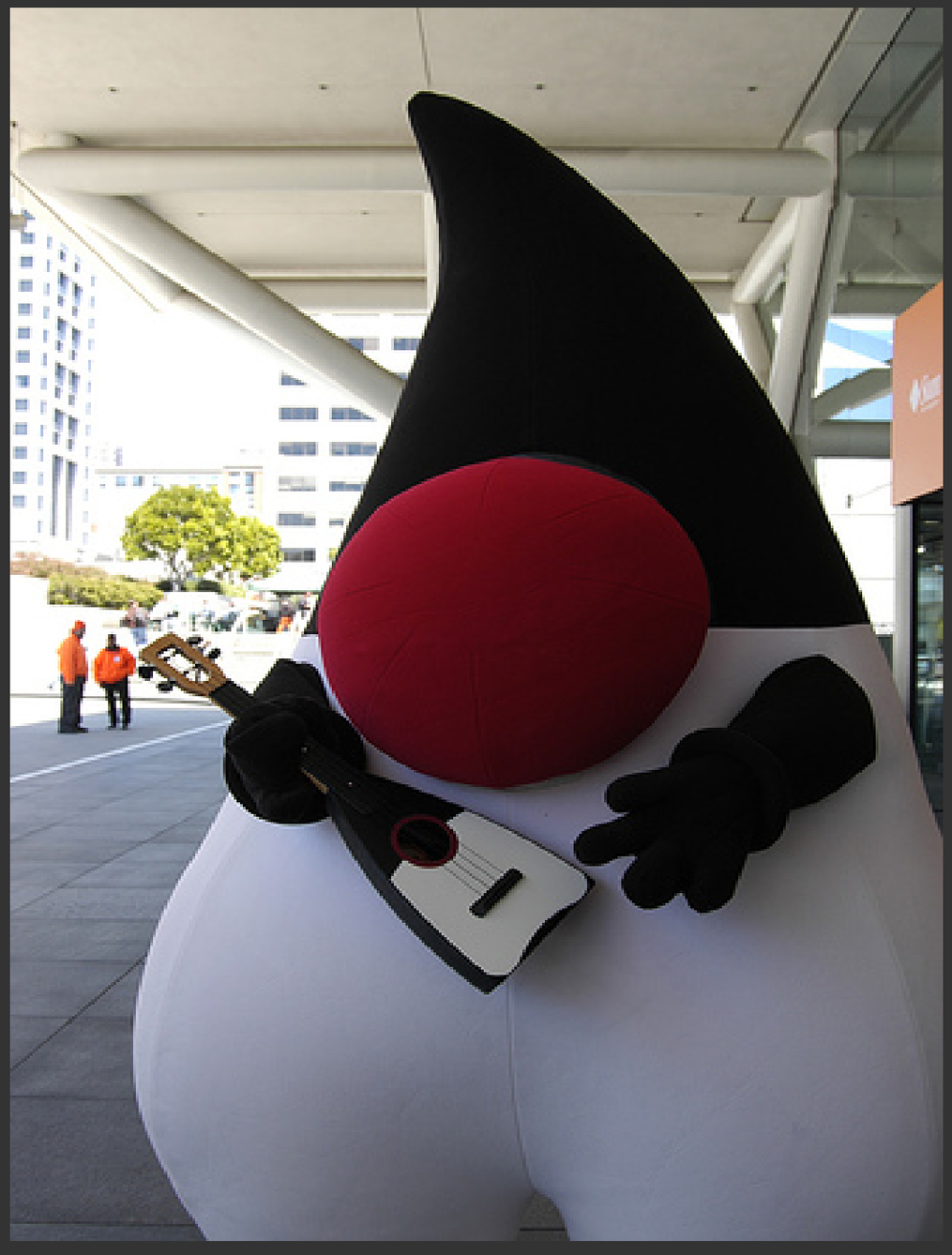
But first ...

Making the connection

... find your happy place ...

Scott Beale / Laughing Squid

http://www.flickr.com/photos/nicholas_t/281820290/sizes/m/



yuichi.sakuraba http://www.flickr.com/photos/skrb/150052412/sizes/m/

We *like* Java, right?

yuichi.sakuraba http://www.flickr.com/photos/skrb/150052412/sizes/m/

Awesome JRuby is Awesome

Fast

Many many many Java libraries

Easy cross-platform GUI development

Picking a Wii lib

WiiRemoteJ

Closed source, jar-only

Web pages spam-littered wiki

Vague license

Java 1.5 or higher

Picking a Wii lib

WiiUseJ

Open source

Hosted at Google Code

GPL v3

Java 1.6

Sits on wiiuse C library

Picking a Wii lib

WiiRemoteJRuby

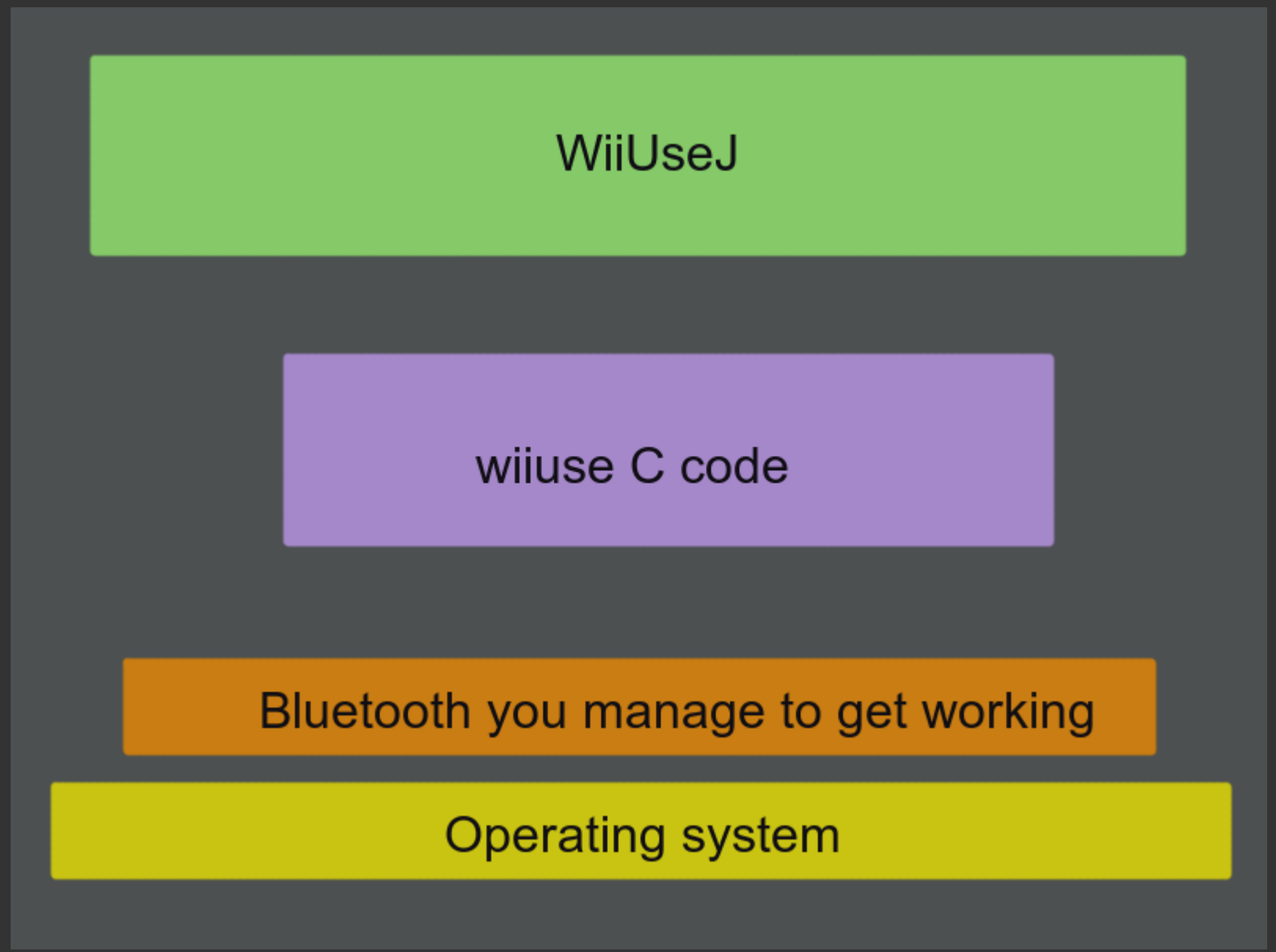
WiiUseJRuby

http://gitorious.org/users/neurogami

Bluetooth

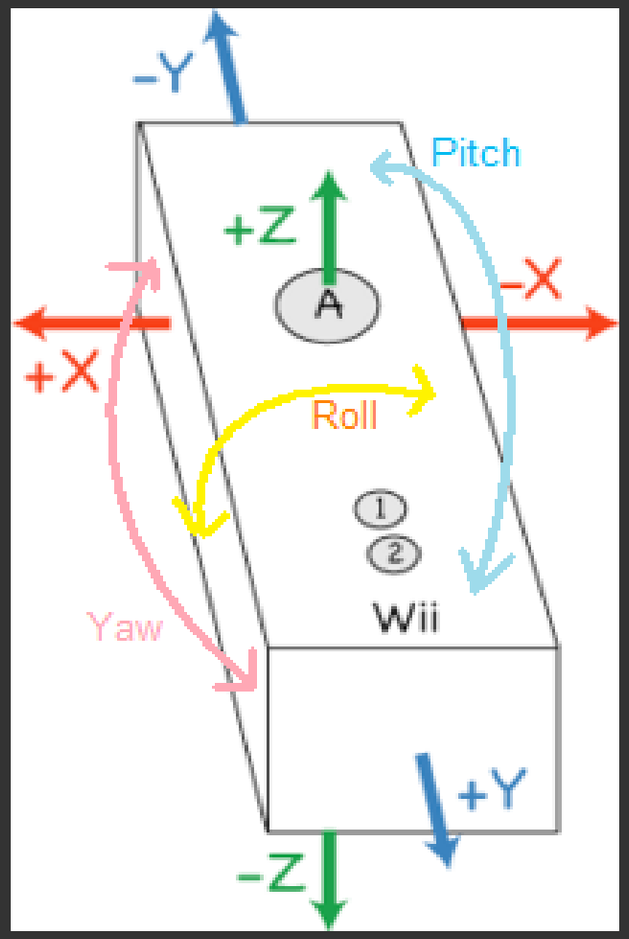
BYOB

Bluetooth


WiiUseJ
wiiuse C code
Bluetooth you manage to get working
Operating system

Wii data basics

Wii data basics

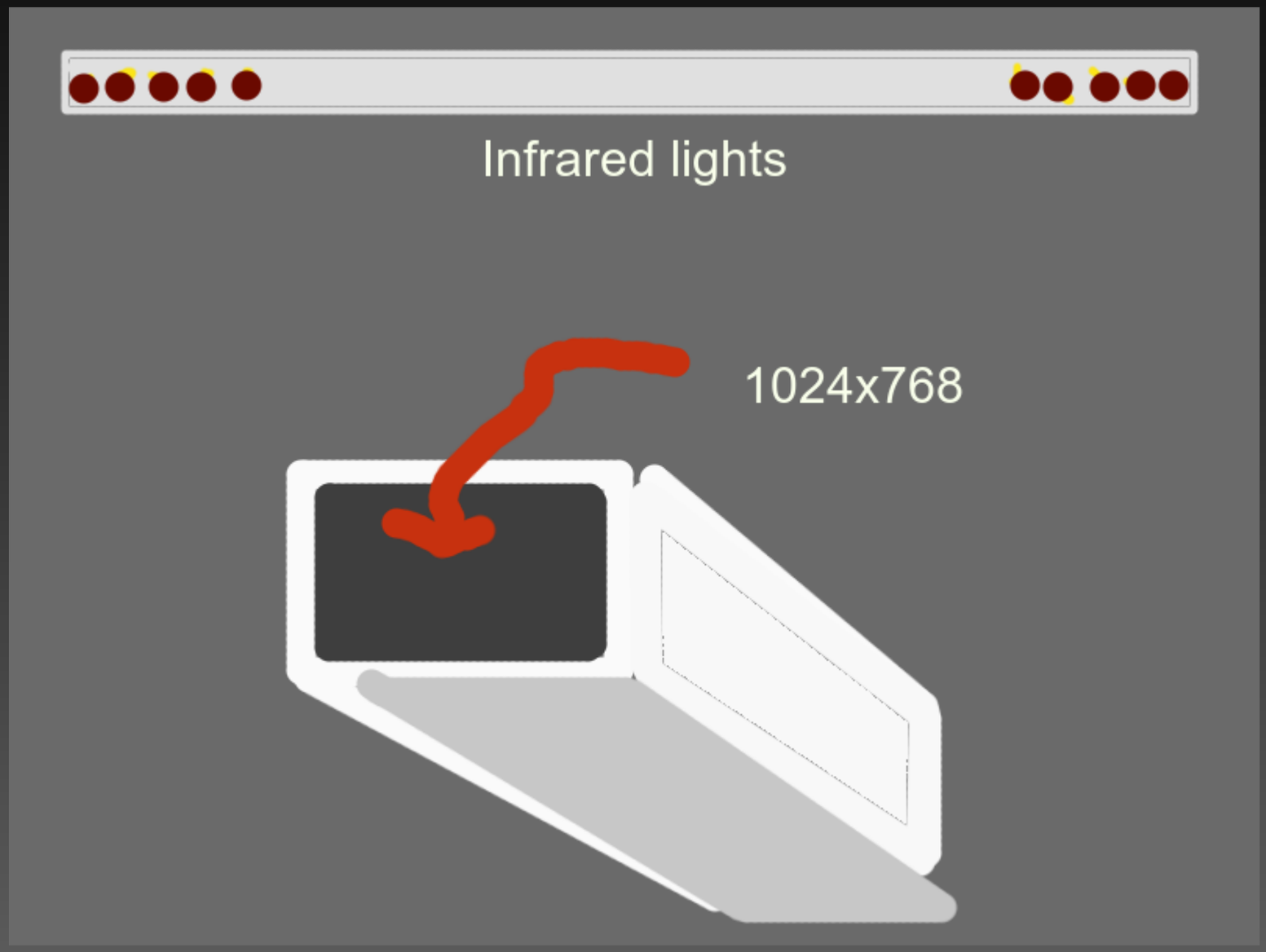


Motion sensing
3-axis accelerometer

X, Y, Z are “flat” motions
Pitch, yaw, roll are rotations

http://www.wiili.org/index.php/Motion_analysis

Wii data basics



Wii data basics

Served up kinda raw

http://www.flickr.com/photos/nikonvscanon/2401210342/sizes/m/

Wii data basics

But we've got Ruby sauce!

http://www.flickr.com/photos/nikonvscanon/2401210342/sizes/m/

The Plan

JRuby so we can use Java libs

Wrap those libs in Ruby

Add a nicer API

Mix with rapid GUI development sweetness

Play

Code

GUI app that reads motion sensor data

Displays values

Makes noise

Monkeybars

monkeybars.org

Wraps Swing with MVC

Watch the RubyConf2008 video

Monkeybars in Nutshell

Controller: handles events

View: Maps UI to model

Model: holds data

Event handling

Swing stuff

okButtonActionPerformed

```
# Your Monkeybars controller

def ok_button_action_performed
  # Cool code here
end
```

Monkeybars stuff

WiiUseJ in Nutshell

Grabs sensor data

Raises events

WiiUseJ in Nutshell

Map to controller methods

WiiUseJRuby in Nutshell

onButtonsEvent

WiiUseJ stuff

Mapping of Wii event to Ruby code

```
# Your Monkeybars controller

def ok_button_action_performed
  # Cool code here
end
```

WiiUseJRuby

```
# Add these in to Wii-enable this controller:
require 'wii_api_manager'
require 'wiimotable'
require 'wiimote_event_listener'

require 'midi'

class DisplayController < ApplicationController
  #  do some Monkeybars stuff  then
  # mixin Wii module ...
  include Neurogami::Wiimotable

  def load
    # Map Wii events to code
   mappings = {
     :wiimote_button_two   => lambda {|e| exit_button_action_performed e },
     :wiimote_button_b     => lambda {|e| b_button_action_performed e },
     :wiimote_button_up    => lambda {|e| increment_set_midi_index },
  # ... more mappings of Wii events to app code
     :motion_sensing_event => lambda{|e|  motion_sensing_event_action_performed e  }
   }

   # Create a Wiimote event listner, passing in the mappings
   wiimote_me Neurogami::WiimoteEventListener.new(mappings)
   @last_motion_event = nil
   @last_ir_event = nil
   init_midi
 end
```

```
# Event handler we've mapped to :motion_sensing_event.
# event has the raw data from the Wii motion event
 def motion_sensing_event_action_performed event

    model.pitch                = event.orientation.pitch
    model.yaw                  = event.orientation.yaw
    model.roll                      = event.orientation.roll
    model.gforce_x                  = event.gforce.x
    model.gforce_y                  = event.gforce.y
    model.gforce_z                  = event.gforce.z
    model.raw_acceleration_x = event.raw_acceleration.x
    model.raw_acceleration_y = event.raw_acceleration.y
    model.raw_acceleration_z = event.raw_acceleration.z

   # Tell the view to render the current model data
   # See monkeybars.org for how this works.
    update_view

  end
```

```
module Neurogami

  class WiimoteEventListener

    WIIMOTE_BUTTON_TWO = 0x0001
    WIIMOTE_BUTTON_ONE = 0x0002
    WIIMOTE_BUTTON_B = 0x0004
    WIIMOTE_BUTTON_A = 0x0008
    WIIMOTE_BUTTON_MINUS = 0x0010
    WIIMOTE_BUTTON_ZACCEL_BIT6 = 0x0020
    WIIMOTE_BUTTON_ZACCEL_BIT7 = 0x0040
    WIIMOTE_BUTTON_HOME = 0x0080
    WIIMOTE_BUTTON_LEFT = 0x0100
    WIIMOTE_BUTTON_RIGHT = 0x0200
    WIIMOTE_BUTTON_DOWN = 0x0400
    WIIMOTE_BUTTON_UP = 0x0800
    WIIMOTE_BUTTON_PLUS = 0x1000
    WIIMOTE_BUTTON_ZACCEL_BIT4 = 0x2000
    WIIMOTE_BUTTON_ZACCEL_BIT5 = 0x4000
    WIIMOTE_BUTTON_UNKNOWN = 0x8000
    WIIMOTE_BUTTON_ALL = 0x1F9F


    def initialize mapping
      define_on_ir_event(mapping.delete :ir_event) if mapping[:ir_event ]
      define_on_motion_sensing_event(mapping.delete :motion_sensing_event)  if
          mapping[:motion_sensing_event ]
      assign_button_mapping mapping
    end

    def define_on_ir_event prk
      self.class.send :define_method, :on_ir_event do |event|
        prk.call(event)
      end
    end

    def define_on_motion_sensing_event prk
      self.class.send :define_method, :on_motion_sensing_event do  |event|
        prk.call(event)
      end
    end


    def assign_button_mapping  mapping
      @button_handlers ||= { }
      mapping.each do |button_key, response|
        key_str = button_key.to_s
        keys = key_str.split( '_and_')
        key_sum = keys.inject(0){|sum, k| sum + eval(k.upcase) }
        @button_handlers[key_sum] = response
      end
    end

    def assign_ir_mapping  prk
      define_method
    end

    #  WiimoteButtonsEvent
    def onButtonsEvent event
      button = event.buttonsJustPressed
      return if button == 0
      return unless handler = @button_handlers[button]
      if handler.respond_to?(:call)
        begin
          handler[event]
        rescue Exception => e
          STDERR.puts " \n ERROR at #{__FILE__}:#{__LINE__}: #{e.inspect} \n handler =
#{handler.pretty_inspect}"
          raise  e
        end
      else
        send handler, event
      end
    end

    def on_ir_event event
    end

    def on_motion_sensing_event event
    end

    def  onExpansionEvent event
    end

    def  onStatusEvent event
    end

    def  onDisconnectionEvent event
    end

    def  onNunchukInsertedEvent event
    end

    def  onNunchukRemovedEvent event
    end
  end
end
```

C'mon James, demo something else

Wii IR Plotting

C'mon James, demo something

Wii AccelerMidi

What's good

Mapping events: Easy

GUI: Easy

MIDI: OK

Play score: 70%

The_Pug_Father http://www.flickr.com/photos/fleur-design/458562896/sizes/m/

What's bad

Precise control: Hard

Project creation: Tedious

Little annoyances: Many

nyki_m http://www.flickr.com/photos/nyki_m/2912470866/sizes/m/

Sadness score: 70%

Making code playable

Easy to get going

Easy to do the obvious

Easy to do the impulsive

Encourage “What if?”

Making code playable

Decoupled

Bite-sized

Inviting

What next

Keep playing

Evolve the API

Improve packaging/set-up

Odds and Ends

Johnny Lee – johnnylee.net/projects/wii/

- Great Wii hacks

Arun Mehta – skid.org.in

- Help kids with cerebral palsy and autism

Code and slides and stuff

- jamesbritt.com
- james.britt@gmail.com
- @jamesbritt

Thanks

Guilhem Duche for WiiUseJ

Charles Nutter, Tom Enobo, and the JRuby team

David Koontz, Logan Barnett for Monkeybars joy

Pat, Mike, and the MWRC team who rock, as usual

All you Ruby hackers

Go play