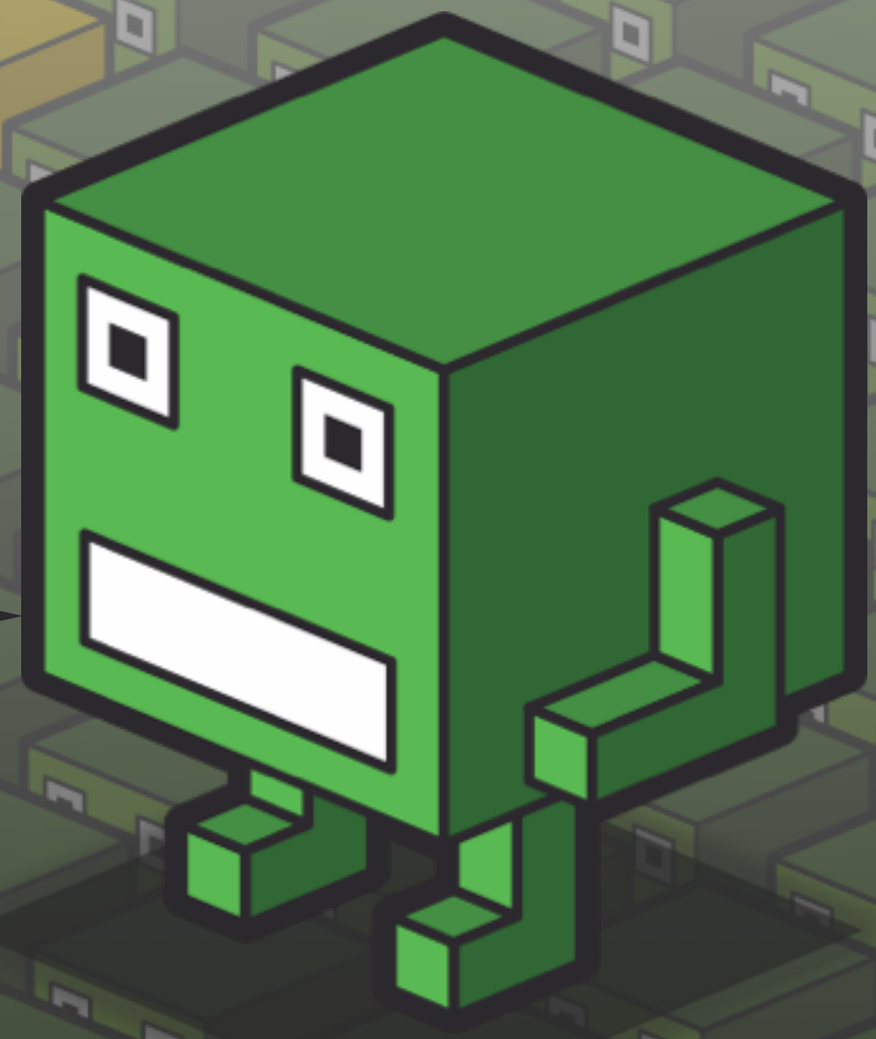
Making Burgers with JavaScript

November 11, 2010



CODEBITS IV

PAVILHÃO ATLÂNTICO 11 - 13 NOVEMBER 2010

Who am I

Who am I

- Diogo Antunes

Who am I

- Diogo Antunes
- JavaScript developer @ SAPO

Who am I

- Diogo Antunes
- JavaScript developer @ SAPO
- @dicode

Who am I

- Diogo Antunes
- JavaScript developer @ SAPO
- @dicode
- http://js.sapo.pt

Should I stay or should I go?

Should I stay or should I go?

- we are going to talk about JavaScript

Should I stay or should I go?

- we are going to talk about JavaScript
- so I will skip some steps

Should I stay or should I go?

- we are going to talk about JavaScript
- so I will skip some steps
- but feel free to bug me if you need

don’t like JavaScript????

Should I stay or should I go?

Saturday, November 13, 2010

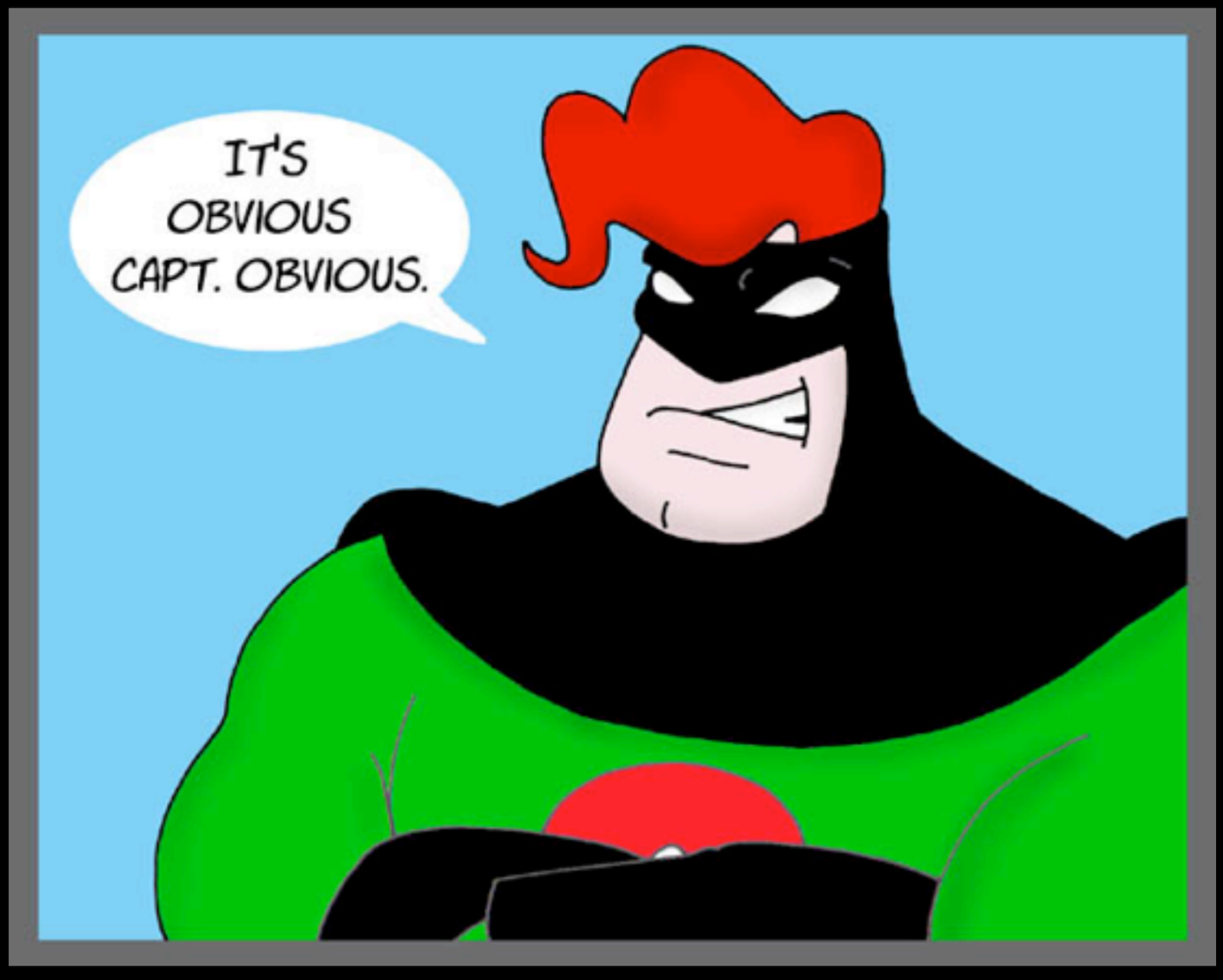
GO AWAY!!!

http://images.icanhascheezburger.com/completestore/2008/12/22/128744751250625583.jpg

What are we talking about?

What are we talking about?

- JavaScript



thank you captain obvious
http://www.eiaonline.com/uploaded_images/captobvious-738633-747223.jpg

What are we talking about?

- JavaScript
- Raphaël - JavaScript Library

What are we talking about?

- JavaScript
- Raphaël - JavaScript Library
- LibSAPO.js - for event handling and stuff

What are we talking about?

- JavaScript
- Raphaël - JavaScript Library
- LibSAPO.js - for event handling and stuff
- and more...

if we have time... basically

Raphaël - wtf?

http://chericos.org/system/files/images/raphael_logo.jpg

Raphaël - wtf?

- JavaScript library to simplify work with vector graphics

Raphaël - wtf?

- JavaScript library to simplify work with vector graphics
- uses SVG W3C recommendation and VML

Raphaël - wtf?

- JavaScript library to simplify work with vector graphics
- uses SVG W3C recommendation and VML
- so every graphic element is a DOM Node

Raphaël - wtf?

- JavaScript library to simplify work with vector graphics
- uses SVG W3C recommendation and VML
- so every graphic element is a DOM Node
- so you can add events to it... YAY!

Raphaël - wtf?

- JavaScript library to simplify work with vector graphics
- uses SVG W3C recommendation and VML
- so every graphic element is a DOM Node
- so you can add events to it... YAY!
- supports FF3+ SF3+ Ch5+ Op9.5+ IE6+

Raphaël - why?

Raphaël - why?

- lightweight - 20k

Raphaël - why?

- lightweight - 20k
- cross browser support

Raphaël - why?

- lightweight - 20k
- cross browser support
- it's easy to work with

Raphaël - why?

- lightweight - 20k
- cross browser support
- it’s easy to work with
- made by Dmitry Baranovskiy who totally ROCKS!!!

Raphaël - how?

Raphaël - how?

- download it

Raphaël - how?

- download it
- include it in your html

```
<script type="text/javascript" src="raphael.js"></script>
```

Raphaël - how?

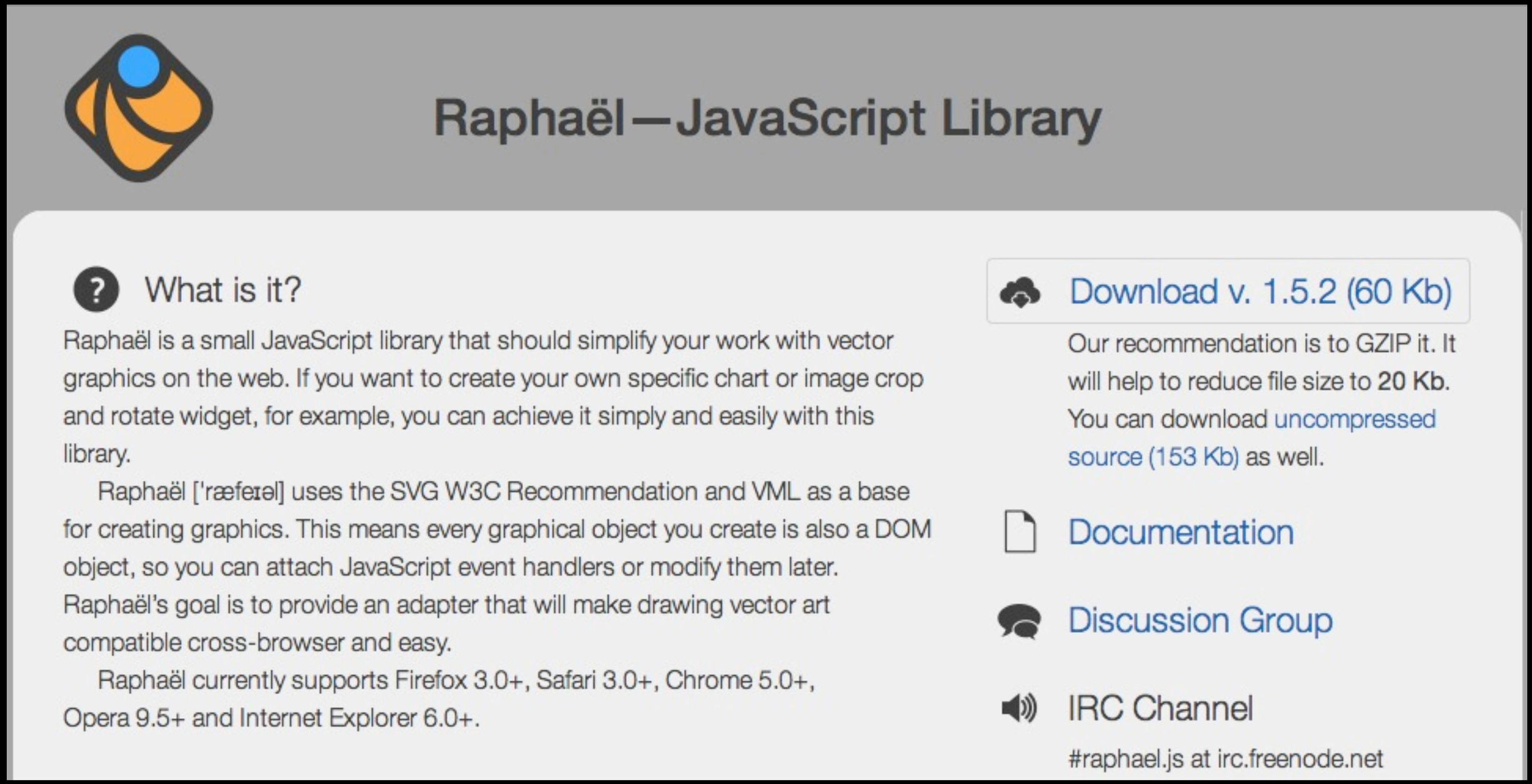
- download it
- include it in your html

```
<script type="text/javascript" src="raphael.js"></script>
```

- and code it!

So in your script tag you can use Raphael
but wait...
what does this really do? and have

Raphaël - demos



You can check the website and see

Raphaël - demos

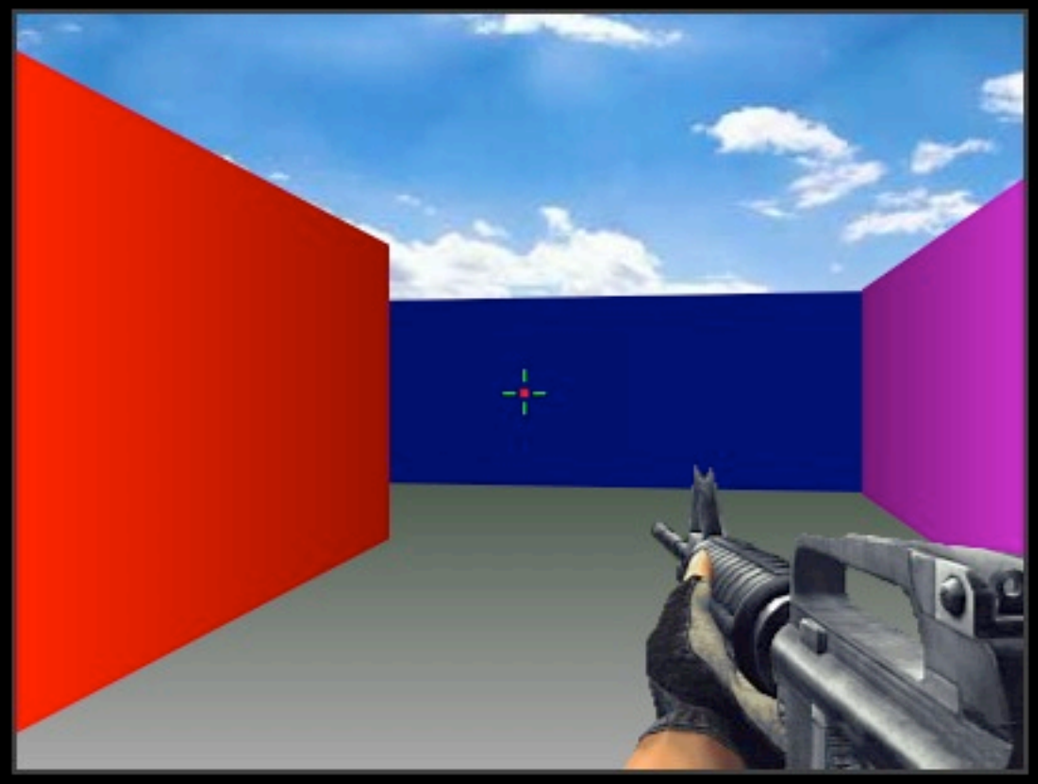
a tiger

Raphaël - demos

text rendering - helvetica example

Raphaël - demos

RaphaelScape – "Maze Mod with Raphaël"



© 2009 Benjamin Joffe
http://www.benjoffe.com/

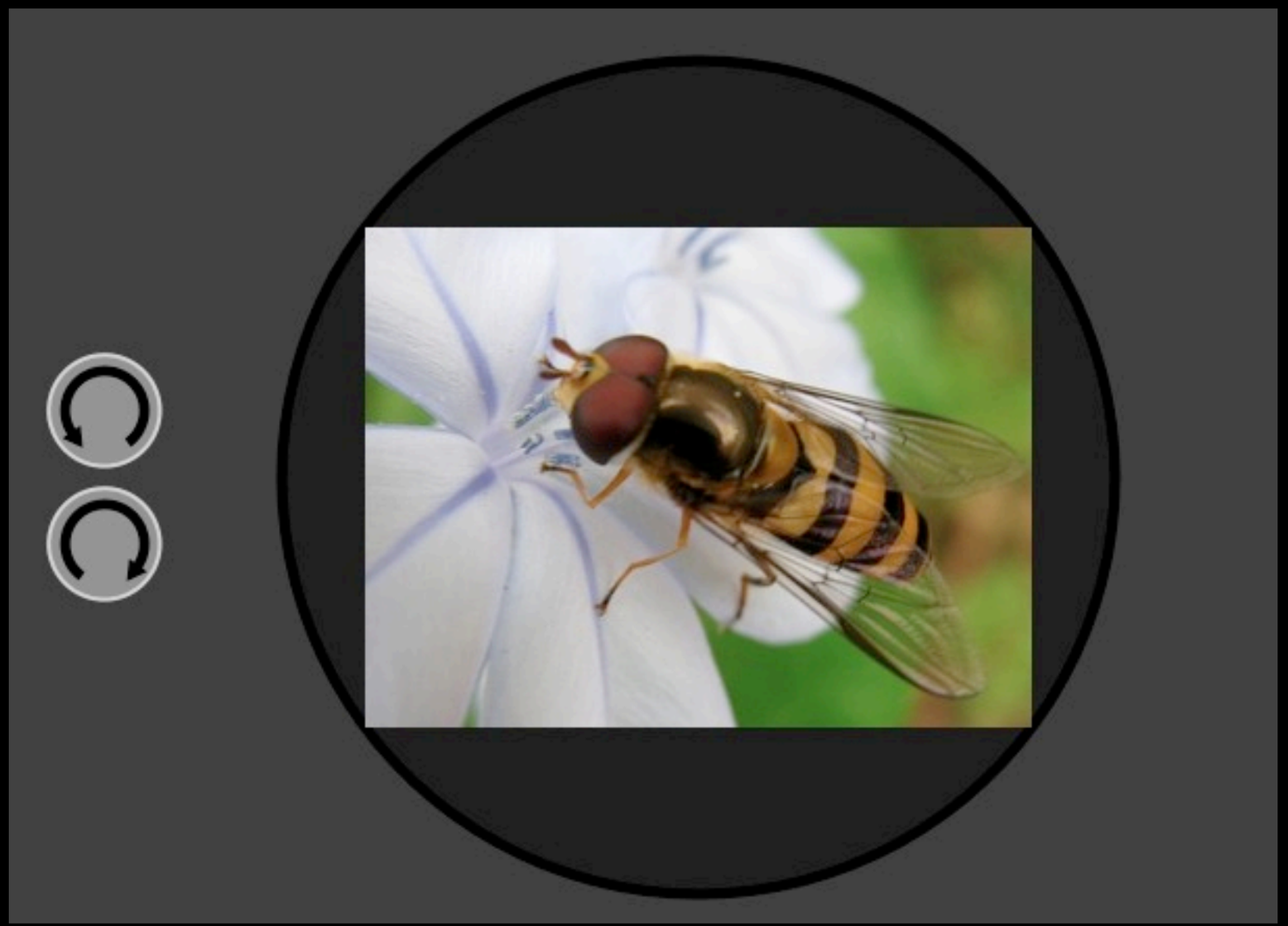
Maze mod and Raphaël conversion
© David A. Faught, 2008–2010
dave.faught@gmail.com

Optimisation of Raphaël code
© Dmitry Baranovskiy, 2010
dmitry@baranovskiy.com

Use the arrow keys to walk around the map. Space bar = jump.

a maze game

Raphaël - demos



how to rotate images

Raphaël - demos

well, event the logo is vector based

Raphaël - the basics

- creating your canvas

Raphaël - the basics

- creating your canvas

```
var canvas = Raphael("container", 400, 300);
```


creating a canvas on element with id container, giving it width 400 and height 300

Raphaël - the basics

- creating your canvas

```
var canvas = Raphael("container", 400, 300);
var canvas = Raphael(container, 400, 300);
```


creating a canvas on element container (previously selected or created), giving it width 400 and height 300

Raphaël - the basics

- creating your canvas

```
var canvas = Raphael("container", 400, 300);
var canvas = Raphael(container, 400, 300);
var canvas = Raphael(10, 10, 400, 300);
```


creating a canvas on top 10 and left 10 of your page giving it width 400 and height 300

Raphaël - the primitives

```
var paper = Raphael(10,10, 500, 500);

paper.circle(100, 100, 50);
```


circle args (xcenter, ycenter, radius)

Raphaël - the primitives

```
var paper = Raphael(10,10, 500, 500);

paper.rect(200, 200, 100, 50);
```


rect args (x, y, width, height)

Raphaël - the primitives

```
var paper = Raphael(10,10, 500, 500);

paper.rect(300, 300, 100, 50, 5);
```


rect with rounded corners args (x, y, width, height, radius)

Raphaël - the primitives

```
var paper = Raphael(10,10, 500, 500);

paper.ellipse(400, 400, 40, 20);
```


ellipse args (xcenter, ycenter, hradius, vradius)

Raphaël - see it happen

```
var paper = Raphael(10, 10, 500, 500);

var path = paper.path("M20 20L245 245");
```


create a path - move to 20, 20 Line Till 145, 145

Raphaël - see it happen

```
var paper = Raphael(10, 10, 500, 500);

var path = paper.path("M20 20L245 245");

var square = paper.rect(15, 15, 10, 10);
```


create a rectangule

Raphaël - see it happen

```
var paper = Raphael(10, 10, 500, 500);

var path = paper.path("M20 20L245 245");

var square = paper.rect(15, 15, 10, 10);

square.animateAlong(path, 4000);
```


make the rectangule animate through the path

Raphaël - see it happen

```
var paper = Raphael(10, 10, 500, 500);

var path = paper.path("M20 20L245 245");

var square = paper.rect(15, 15, 10, 10);

square.animateAlong(path, 4000, function(){

    setInterval(function(){
        square.rotate(1);
    }, 10);

});
```


and animate rotation in the end...

Raphaël - text

- you can use raphael with cufon to enhance your site look

you can use raphael with cufon to enhance your site look - do not abuse!!!

Raphaël - demo svg to js :)

- so you use some graphic sw to make an svg

like gimp or illustrator

Raphaël - demo svg to js :)

- so you use some graphic sw to make an svg
- then you want to use it with Raphaël

Raphaël - demo svg to js :)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 15.0.1, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_2" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
	 width="296px" height="86.811px" viewBox="0 0 296 86.811" enable-background="new 0 0 296 86.811" xml:space="preserve">
<linearGradient id="SVGID_1_" gradientUnits="userSpaceOnUse" x1="52.7769" y1="-4.7632" x2="271.7753" y2="83.2362">
	<stop  offset="0" style="stop-color:#FBB31B"/>
	<stop  offset="1" style="stop-color:#FDD858"/>
</linearGradient>
<path fill="url(#SVGID_1_)" d="M6.767,10.197c0,0,1.997,14.082,20.844,19.404c3.484,1.108,20.097,5.415,25.104,9.688
	c0.44,0.134,17.162,10.868,20.787,14.432s14.347,11.632,19.921,16.735s11.465,12.482,12.739,12.85s3.293,0.924,4.941,0
	s4.771-4.134,6.861-6.232s10.98-7.547,13.433-8.96s10.315-5.903,13.327-7.751s17.265-11.646,21.045-15.457
	c3.192-1.106,6.315-2.958,8.493-3.347s25.459-4.583,27.804-1.907c3.803-0.711,9.068-2.409,10.987-3.34s6.605-3.521,6.605-3.521
	s7.754-3.467,10.855-3.186c3.104,0.282,6.045,1.998,14.852-1.344c8.809-3.342,14.371-9.2,19.752-7.401s17.279,10.287,15.77,21.276
	c0.26,2.564-1.074,11.543,0.408,11.562c1.48,0.02,8.861-0.249,8.699-2.838s-3.24-16.284-6.578-20.747
	c-3.336-4.463-9.035-13.457-17.533-15.738c-2.246-0.005-87.95-1.256-104.335-4.177S71.715,1.003,59.395,2.384
	c-3.461,0.802-22.562-0.335-32.25,2.391S6.767,10.197,6.767,10.197z"/>
</svg>
```


when you edit and svg file in a text editor you get

Raphaël - demo svg to js :)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 15.0.1, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_2" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
	 width="296px" height="86.811px" viewBox="0 0 296 86.811" enable-background="new 0 0 296 86.811" xml:space="preserve">
<linearGradient id="SVGID_1_" gradientUnits="userSpaceOnUse" x1="52.7769" y1="-4.7632" x2="271.7753" y2="83.2362">
	<stop  offset="0" style="stop-color:#FBB31B"/>
	<stop  offset="1" style="stop-color:#FDD858"/>
</linearGradient>
<path fill="url(#SVGID_1_)" d="M6.767,10.197c0,0,1.997,14.082,20.844,19.404c3.484,1.108,20.097,5.415,25.104,9.688
	c0.44,0.134,17.162,10.868,20.787,14.432s14.347,11.632,19.921,16.735s11.465,12.482,12.739,12.85s3.293,0.924,4.941,0
	s4.771-4.134,6.861-6.232s10.98-7.547,13.433-8.96s10.315-5.903,13.327-7.751s17.265-11.646,21.045-15.457
	c3.192-1.106,6.315-2.958,8.493-3.347s25.459-4.583,27.804-1.907c3.803-0.711,9.068-2.409,10.987-3.34s6.605-3.521,6.605-3.521
	s7.754-3.467,10.855-3.186c3.104,0.282,6.045,1.998,14.852-1.344c8.809-3.342,14.371-9.2,19.752-7.401s17.279,10.287,15.77,21.276
	c0.26,2.564-1.074,11.543,0.408,11.562c1.48,0.02,8.861-0.249,8.699-2.838s-3.24-16.284-6.578-20.747
	c-3.336-4.463-9.035-13.457-17.533-15.738c-2.246-0.005-87.95-1.256-104.335-4.177S71.715,1.003,59.395,2.384
	c-3.461,0.802-22.562-0.335-32.25,2.391S6.767,10.197,6.767,10.197z"/>
</svg>
```


you have path info for raphaël use

Raphaël - demo svg to js :)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 15.0.1, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_2" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
	 width="296px" height="86.811px" viewBox="0 0 296 86.811" enable-background="new 0 0 296 86.811" xml:space="preserve">
<linearGradient id="SVGID_1_" gradientUnits="userSpaceOnUse" x1="52.7769" y1="-4.7632" x2="271.7753" y2="83.2362">
	<stop  offset="0" style="stop-color:#FBB31B"/>
	<stop  offset="1" style="stop-color:#FDD858"/>
</linearGradient>
<path fill="url(#SVGID_1_)" d="M6.767,10.197c0,0,1.997,14.082,20.844,19.404c3.484,1.108,20.097,5.415,25.104,9.688
	c0.44,0.134,17.162,10.868,20.787,14.432s14.347,11.632,19.921,16.735s11.465,12.482,12.739,12.85s3.293,0.924,4.941,0
	s4.771-4.134,6.861-6.232s10.98-7.547,13.433-8.96s10.315-5.903,13.327-7.751s17.265-11.646,21.045-15.457
	c3.192-1.106,6.315-2.958,8.493-3.347s25.459-4.583,27.804-1.907c3.803-0.711,9.068-2.409,10.987-3.34s6.605-3.521,6.605-3.521
	s7.754-3.467,10.855-3.186c3.104,0.282,6.045,1.998,14.852-1.344c8.809-3.342,14.371-9.2,19.752-7.401s17.279,10.287,15.77,21.276
	c0.26,2.564-1.074,11.543,0.408,11.562c1.48,0.02,8.861-0.249,8.699-2.838s-3.24-16.284-6.578-20.747
	c-3.336-4.463-9.035-13.457-17.533-15.738c-2.246-0.005-87.95-1.256-104.335-4.177S71.715,1.003,59.395,2.384
	c-3.461,0.802-22.562-0.335-32.25,2.391S6.767,10.197,6.767,10.197z"/>
</svg>
```


and you have color info for linear gradient

Raphaël - demo svg to js :)

```
var paper = Raphael(0, 0, 296, 87);
```


get your stage done

Raphaël - demo svg to js :)

```
var paper = Raphael(0, 0, 296, 87);

var cheesePath = paper.path("M6.767,10.197c0,0,1.997,14.082,20.844,19.404c3.484,1.108,20.097,5.415,25.104,9.688"+
   "c0.44,0.134,17.162,10.868,20.787,14.432s14.347,11.632,19.921,16.735s11.465,12.482,12.739,12.85s3.293,0.924,4.941,0"+
   "s4.771-4.134,6.861-6.232s10.98-7.547,13.433-8.96s10.315-5.903,13.327-7.751s17.265-11.646,21.045-15.457"+
   "c3.192-1.106,6.315-2.958,8.493-3.347s25.459-4.583,27.804-1.907c3.803-0.711,9.068-2.409,10.987-3.34s6.605-3.521,6.605-3.521"+
   "s7.754-3.467,10.855-3.186c3.104,0.282,6.045,1.998,14.852-1.344c8.809-3.342,14.371-9.2,19.752-7.401s17.279,10.287,15.77,21.276"+
   "c0.26,2.564-1.074,11.543,0.408,11.562c1.48,0.02,8.861-0.249,8.699-2.838s-3.24-16.284-6.578-20.747"+
   "c-3.336-4.463-9.035-13.457-17.533-15.738c-2.246-0.005-87.95-1.256-104.335-4.177S71.715,1.003,59.395,2.384"+
   "c-3.461,0.802-22.562-0.335-32.25,2.391S6.767,10.197,6.767,10.197z");
```


put your path in it

Raphaël - demo svg to js :)

```
var paper = Raphael(0, 0, 296, 87);

var cheesePath = paper.path("M6.767,10.197c0,0,1.997,14.082,20.844,19.404c3.484,1.108,20.097,5.415,25.104,9.688"+
   "c0.44,0.134,17.162,10.868,20.787,14.432s14.347,11.632,19.921,16.735s11.465,12.482,12.739,12.85s3.293,0.924,4.941,0"+
   "s4.771-4.134,6.861-6.232s10.98-7.547,13.433-8.96s10.315-5.903,13.327-7.751s17.265-11.646,21.045-15.457"+
   "c3.192-1.106,6.315-2.958,8.493-3.347s25.459-4.583,27.804-1.907c3.803-0.711,9.068-2.409,10.987-3.34s6.605-3.521,6.605-3.521"+
   "s7.754-3.467,10.855-3.186c3.104,0.282,6.045,1.998,14.852-1.344c8.809-3.342,14.371-9.2,19.752-7.401s17.279,10.287,15.77,21.276"+
   "c0.26,2.564-1.074,11.543,0.408,11.562c1.48,0.02,8.861-0.249,8.699-2.838s-3.24-16.284-6.578-20.747"+
   "c-3.336-4.463-9.035-13.457-17.533-15.738c-2.246-0.005-87.95-1.256-104.335-4.177S71.715,1.003,59.395,2.384"+
   "c-3.461,0.802-22.562-0.335-32.25,2.391S6.767,10.197,6.767,10.197z");

cheesePath.attr({fill: '0-#FBB31B-#FDD858', stroke: 'none'});
```


add the linear gradient ‹angle›-‹colour›[-‹colour›[:‹offset›]]*-‹colour›
removed the stroke so doesn’t default to black

Raphaël - demo svg to js :)

SVG file

Raphaël

raphaël version rendered in chrome
now make it easy you have a php script that does this

where are the burgers?

where are the burgers?

here they are

where are the burgers?

big deal... i can do that in my sleep
http://zedomax.com/blog/wp-content/uploads/2009/02/burger-bed.jpg

let’s make a game!

http://techau.tv/blog/images/GameOnExhibitionmythoughts_12815/GameOn.jpg

let's make a game!

- using raphaël to manage graphics

let's make a game!

- using raphaël to manage graphics
- and LibSAPO.js to handle events and stuff

LibSAPO.js - wtf?

LibSAPO.js - wtf?

- SAPO JavaScript Library

LibSAPO.js - wtf?

- SAPO JavaScript Library
- previously Prototypejs based

LibSAPO.js - wtf?

- SAPO JavaScript Library
- previously Prototypejs based
- now Prototypejs free :)

LibSAPO.js - wtf?

- SAPO JavaScript Library
- previously Prototypejs based
- now Prototypejs free :)
- module oriented

LibSAPO.js - why?

LibSAPO.js - why?

- oriented to SAPO needs

LibSAPO.js - why?

- oriented to SAPO needs
- feature add on demand

LibSAPO.js - why?

- oriented to SAPO needs
- feature add on demand
- bug solving simpler and quicker

LibSAPO.js - why?

- oriented to SAPO needs
- feature add on demand
- bug solving simpler and quicker
- modular and performance target

to fork other js framework
would be harder to code reuse and recode addons

LibSAPO.js - how?

LibSAPO.js - how?

- all in http://js.sapo.pt/

LibSAPO.js - how?

- all in http://js.sapo.pt/
- include /SAPO/

LibSAPO.js - how?

- all in http://js.sapo.pt/
- include /SAPO/
- and then the module you want

LibSAPO.js - how?

- all in http://js.sapo.pt/
- include /SAPO/
- and then the module you want
- in our case

LibSAPO.js - how?

- all in http://js.sapo.pt/
- include /SAPO/
- and then the module you want
- in our case
- /SAPO/Dom/Event/0.1/

LibSAPO.js - demo

```
   <div id="clickMe">click Me</div>

<script type="text/javascript">

var clickMe = s$('clickMe');

SAPO.Dom.Event.observe(clickMe, 'click', function(){
    alert(this.innerHTML + " - DONE & handled as you like! ");
}.bindObjEvent(clickMe));

</script>
```


this is a simple LibSAPO.js demo for event handling

the game!

the game!

- so you have two slices of bread

the game!

- so you have two slices of bread
- and 3 ingredients

the game!

- so you have two slices of bread
- and 3 ingredients
- now we're making it to other people

the game!

- so you have two slices of bread
- and 3 ingredients
- now we're making it to other people

- so you need to put the ingredients

the game!

- so you have two slices of bread
- and 3 ingredients
- now we're making it to other people

- so you need to put the ingredients
- in the correct amount and order

the game! - ingredients

- making the ingredients with raphaël

the game! - ingredients

- making the ingredients with raphaël
- since they are js objects you can reuse it

the game! - ingredients

- making the ingredients with raphaël
- since they are js objects you can reuse it
- scale it

the game! - ingredients

- making the ingredients with raphaël
- since they are js objects you can reuse it
- scale it
- and use it as many times as you want

the game!

- now lets put the elements in place and make a cute start button

the game!

- Scoreboard

the game!

- Scoreboard
- with right and wrong feedback

the game!

- let’s set the events for start

the game!

- let’s set the events for start
- add the elements to our burger

the game!

- let’s set the events for start
- add the elements to our burger
- let’s put a counter

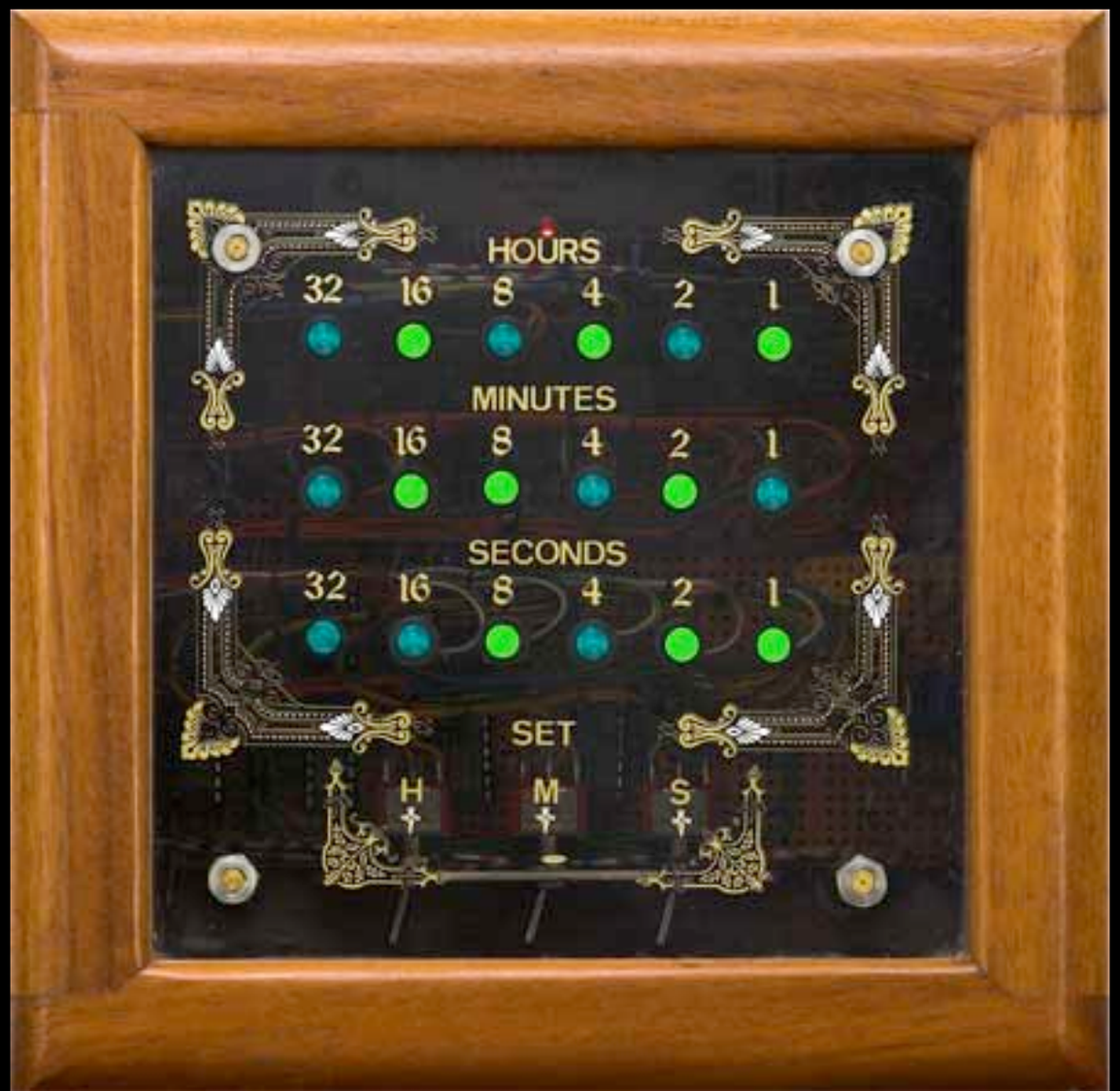
wanna play?

http://common1.csnimages.com/lf/1/hash/2328/2773778/1/Jaxx-Jr-Kids-Foam-Bean-Bag-Gamer-Chair.jpg

wanna play?

- http://dicode.org/codebits2010/makingBurgers.html
- http://6svz.sl.pt

Questions?



Do we still have time?

http://home.btconnect.com/brettoliver1/images/Masterclock/binary_clock.jpg

gRaphaël

partial print screen of http://g.raphaeljs.com/

gRaphaël

- put the goodness into charts!

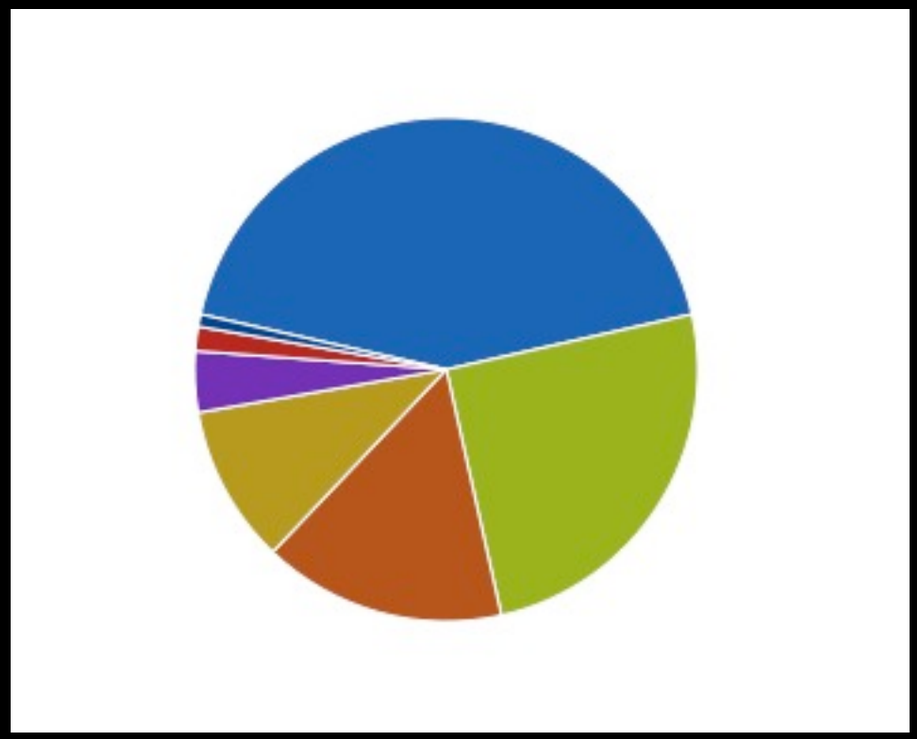
gRaphaël

- put the goodness into charts!
- having raphael.js included

gRaphaël

- put the goodness into charts!
- having raphael.js included
- Download and include g.raphael.js and any (or all) of g.line.js, g.bar.js, g.dot.js and g.pie.js into your HTML page

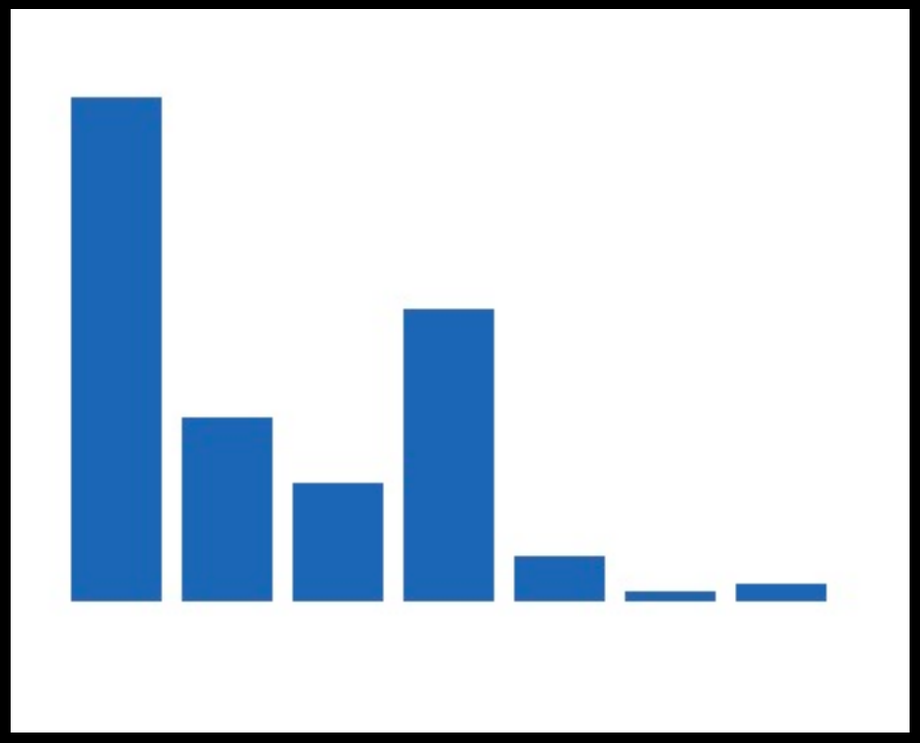
gRaphaël pie chart



```
var r = Raphael(10, 50, 640, 480);
r.g.piechart(320, 240, 100, [55, 20, 13, 32, 5, 1, 2]);
```


xcenter, ycenter, radius, data

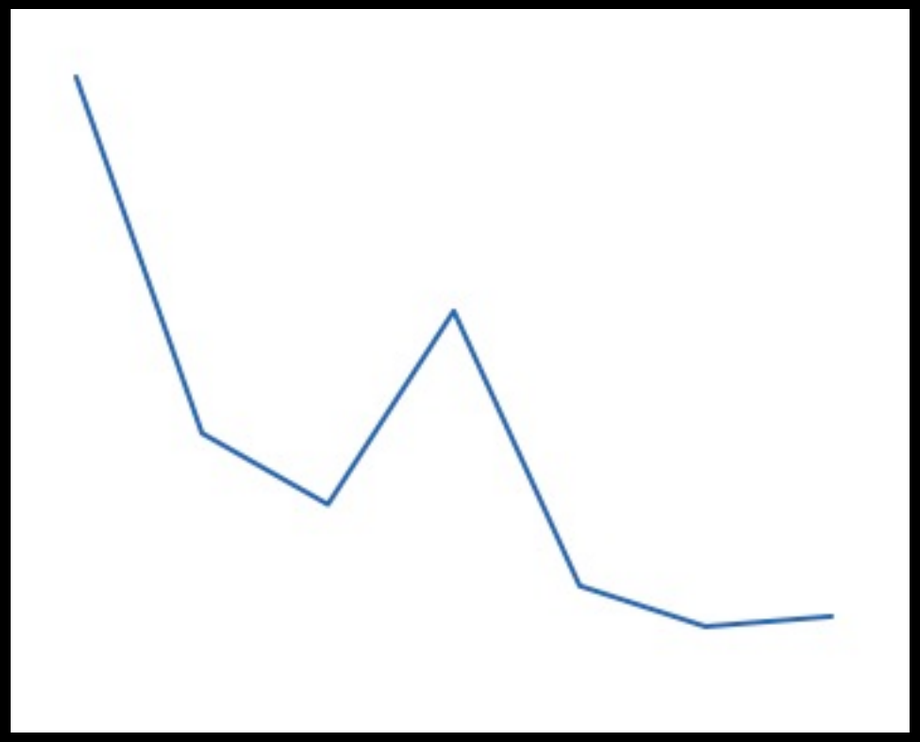
gRaphaël bar chart



```
var r = Raphael(10, 50, 640, 480);
r.g.barchart(10, 10, 320, 240, [[55, 20, 13, 32, 5, 1, 2]]);
```


x, y, w, h, data

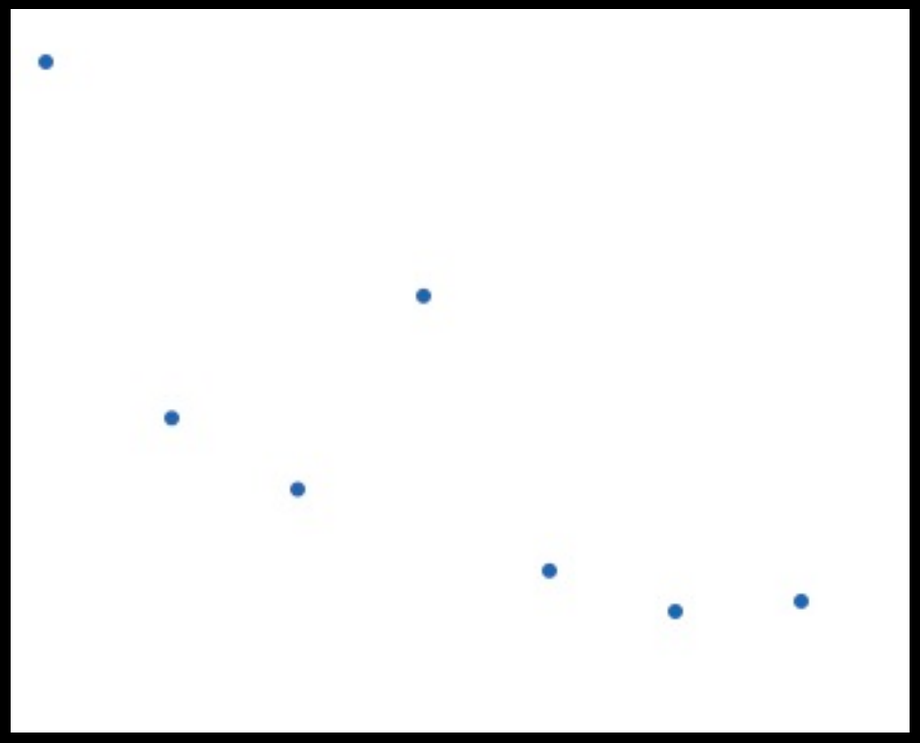
gRaphaël line chart



```
var r = Raphael(10, 50, 640, 480);
r.g.linechart(10, 10, 320, 240, [0, 1, 2, 3, 4, 5, 6],
                [55, 20, 13, 32, 5, 1, 2]);
```


x, y, w, h, xdata, ydata

gRaphaël dot chart



```
var r = Raphael(10, 50, 640, 480);
r.g.dotchart(10, 10, 320, 240, [0, 1, 2, 3, 4, 5, 6],
             [55, 20, 13, 32, 5, 1, 2], [1, 1, 1, 1, 1, 1, 1],
             {max: 3});
```


x, y, w, h, xdata, ydata, caption, opts (in this case dot weight)

that's all folks!

Diogo Antunes
JavaScript developer @ SAPO
twitter: @dicode
email: diogo.j.antunes@co.sapo.pt
im: diogoantunes@sapo.pt

http://js.sapo.pt

http://libsapojs.blogs.sapo.pt

http://dicode.org

Thanks to:

Dmitry Baranovskiy - slides inspiration and examples from the website

Hugo França - graphics for the game

Credits

Please check notes in each slide to check slide notes which reference each images i have used in this presentation