



Tutorial

Pharo-Based TDD for Javascript Apps

Noury Bouraqadi & Dave Mason

<http://pharojs.org>



IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille



Phar JS

=

Develop in Pharo

+

Run on Javascript

Phar JS

MIT
License

=

Framework

+

Tools



Why *Phar*JS ?

- Develop in Smalltalk all the time 😊

- Portability



- Front-end/standalone

- Reuse existing JS libraries

- Run-time Speed



Develop in **Pharo**, Run on **Javascript**

100%

Pharo

0%

Development

Production

0%

Javascript

100%

Develop in **Pharo**, Run on **Javascript**

100%

Pharo

0%

Smalltalk classes

0%

Set of classes +
entry point class

100%

Develop in **Pharo**, Run on **Javascript**

100%

Pharo

0%

Convert Smalltalk to JS

0%

**Only classes reachable
from the
entry point class**

100%

Lifecycle with *Phar*

100%

Pharo

0%

1. Write Tests

2. Pass the tests


3. Export to JS

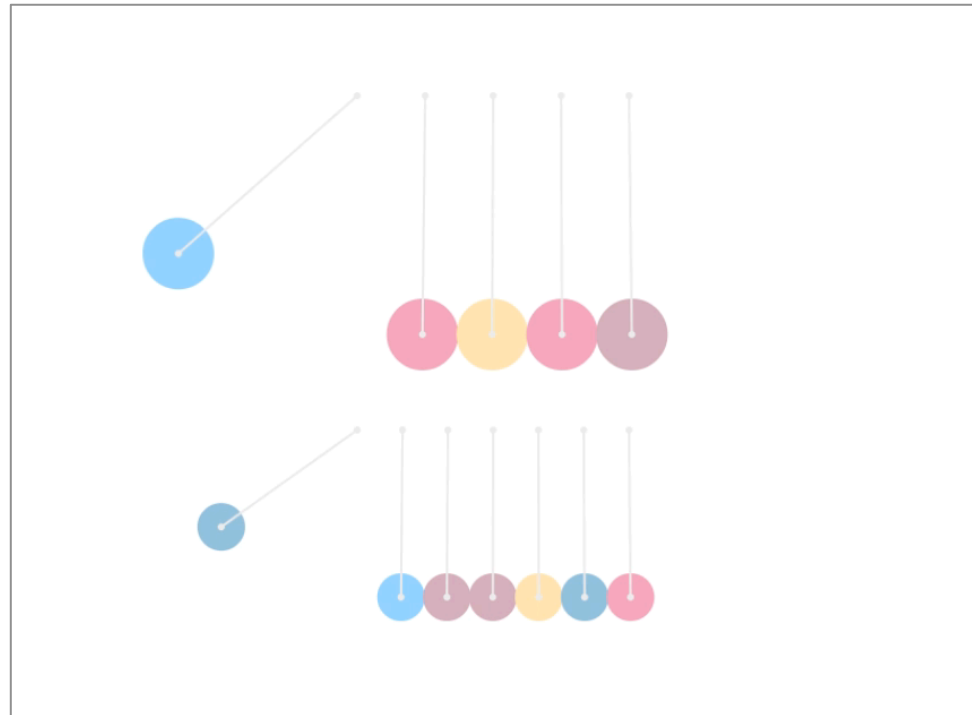
0%

Javascript

100%

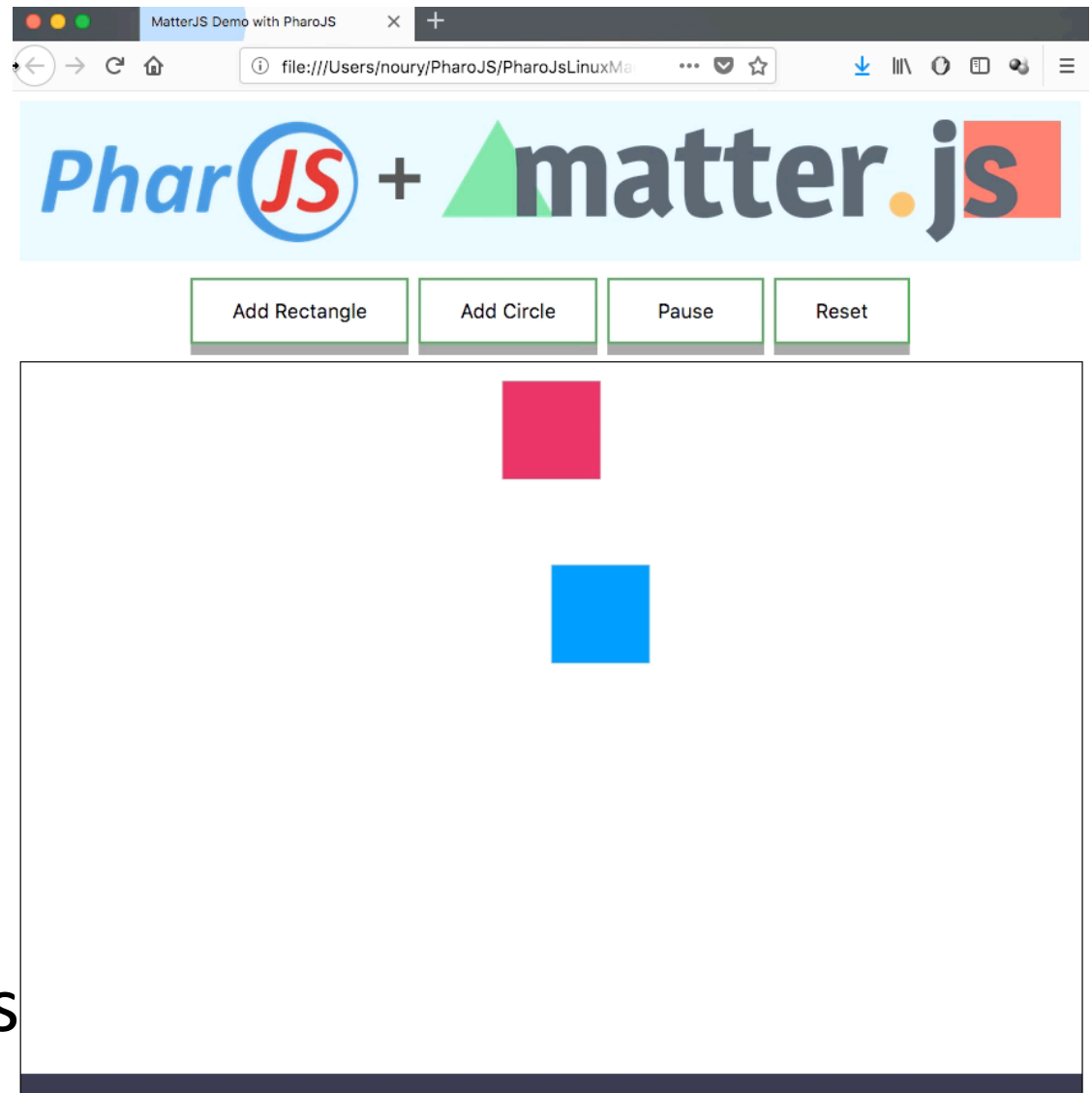
Case Study: TDD of a Physics Simulation

- Reuse an existing JS Library 
- Testing hybrid app = Pharo + Javascript
- Link Pharo code to JS objects including DOM
- Pushing to production



Backlog

- At start up 3 bodies
 - 1 floor
 - 2 falling rectangles
- "Add xxx" buttons
 - Adds a body
- "Reset" button
 - Removes all bodies



Write Tests 100% Pharo & 0% JS


PjWebAppTestCase subclass: #PhysicsSimTest

instanceVariableNames: ''

classVariableNames: ''

package: 'ESUG2018'

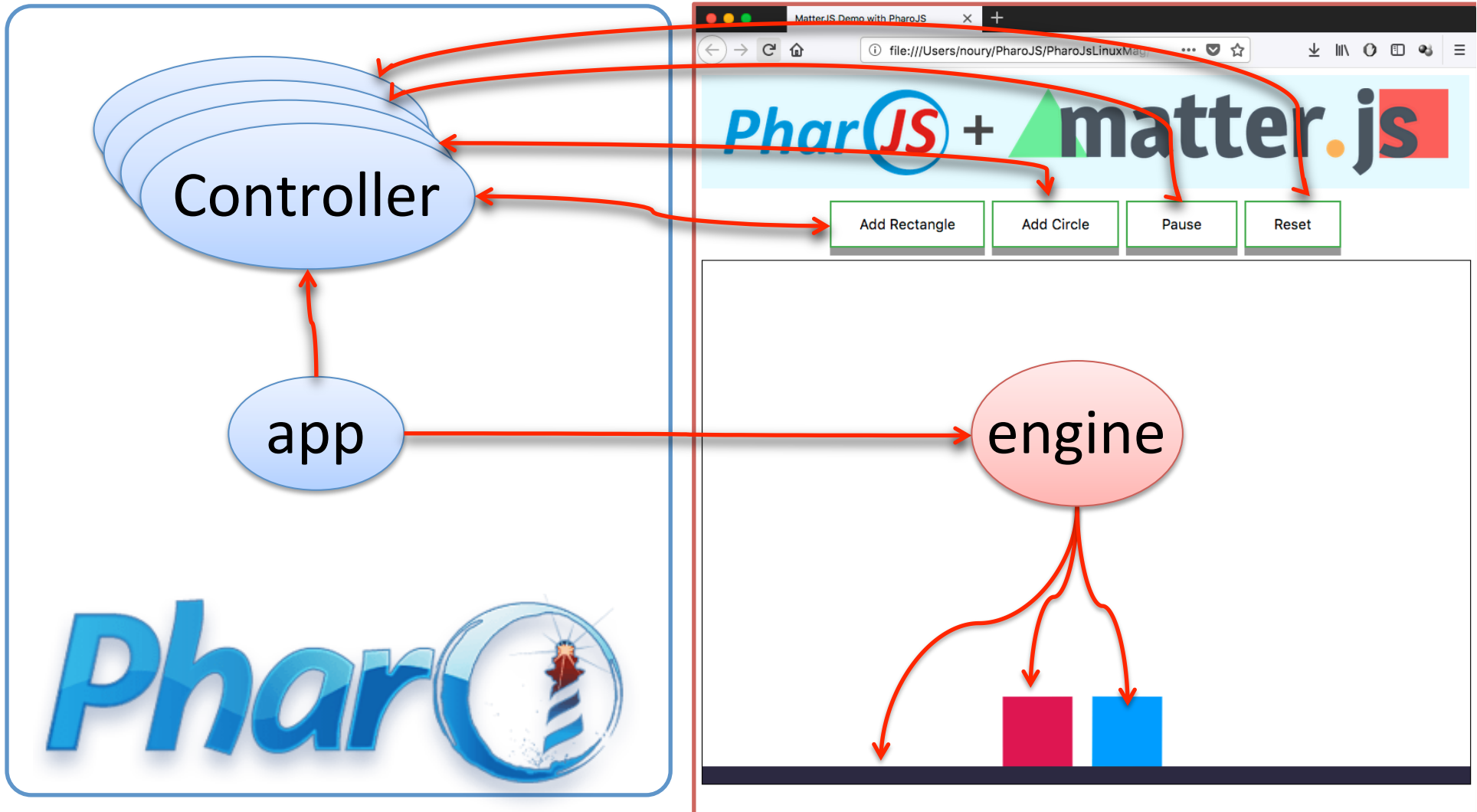
Write Tests 100% Pharo & 0% JS



App should
run on a web
browser

```
PjWebAppTestCase subclass: #PhysicsSimTest  
instanceVariableNames: ''  
classVariableNames: ''  
package: 'ESUG2018'
```

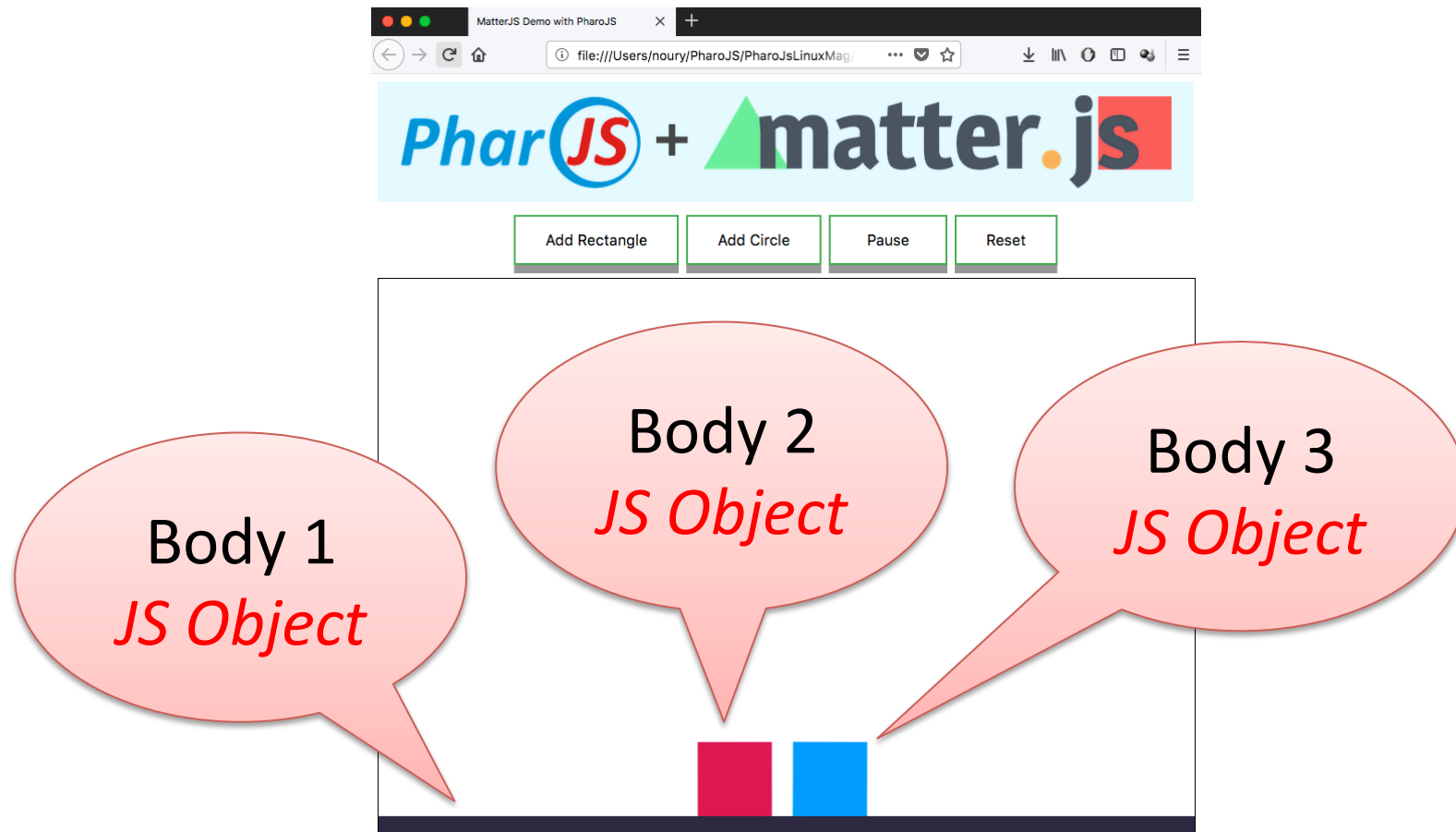
Running the Tests ~50% Pharo & ~50% JS



Running the Tests ~50% Pharo & ~50% JS

PhysicsSimTest>>testAppInitialSetup

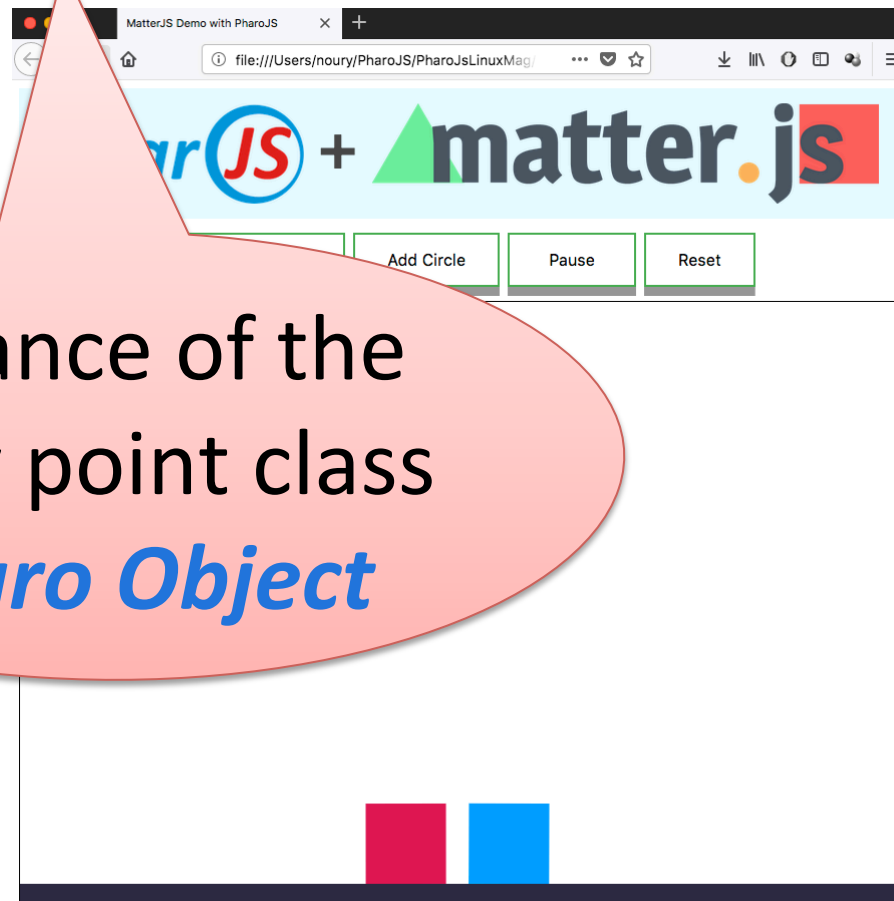
self assert: **app bodies** size equals: **3**.



Running the Tests ~50% Pharo & ~50% JS

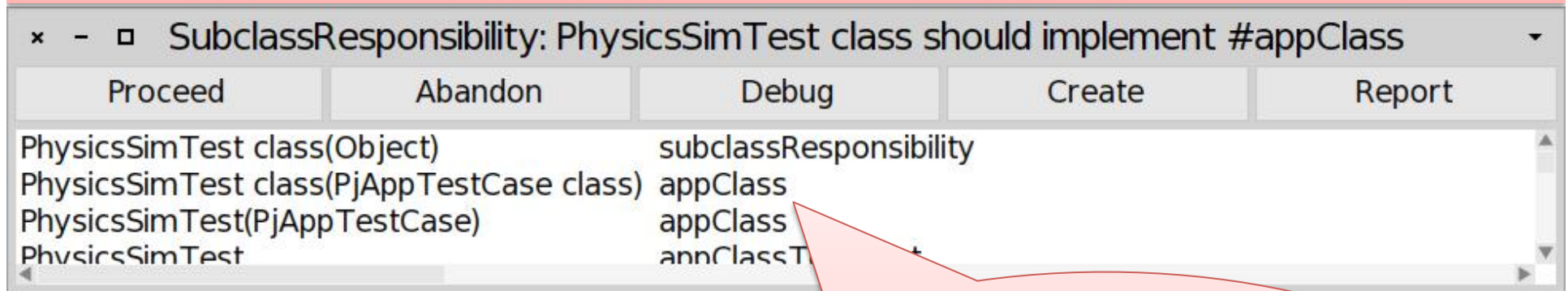
PhysicsSimTest>>testAppInitialSetup

self assert: **app** bodies size equals: 3.



Running the Tests ~50% Pharo & ~50% JS

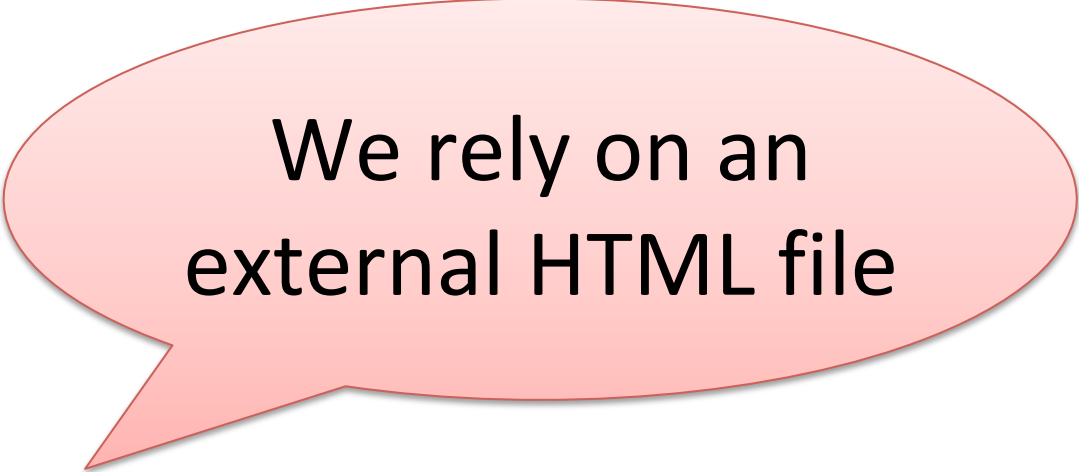
appClass subclassResponsibility



Proceed	Abandon	Debug	Create	Report
PhysicsSimTest class(Object)		subclassResponsibility		
PhysicsSimTest class(PjAppTestCase class)		appClass		
PhysicsSimTest(PjAppTestCase)		appClass		
PhysicsSimTest		appClassT		

Missing
entry point class

Defining the Entry Point Class



We rely on an
external HTML file

```
PjFileBasedWebApp subclass: #PhysicsSim  
instanceVariableNames: 'engine'  
classVariableNames: ''  
package: 'ESUG2018'
```

Test References Entry Point Class

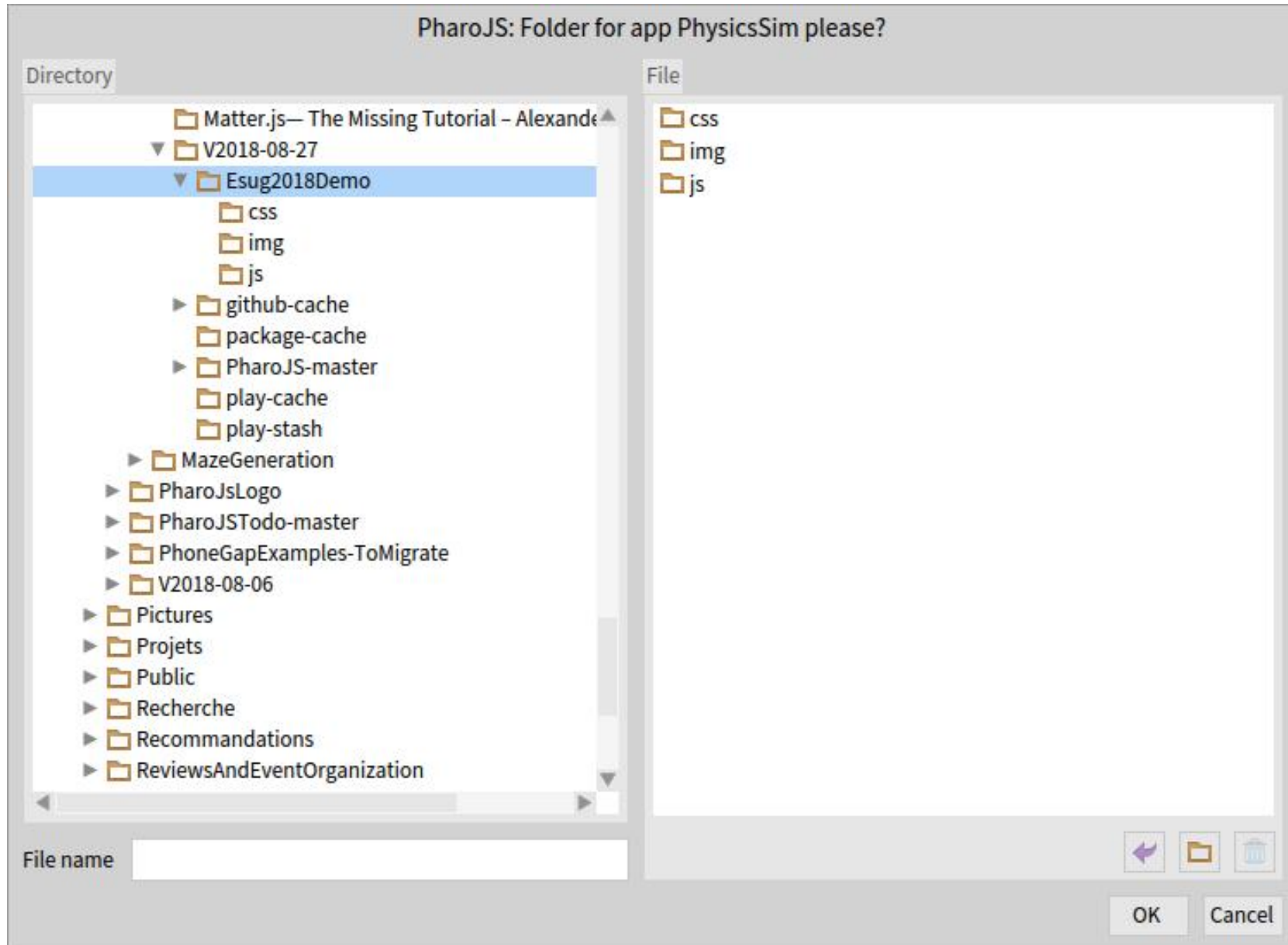
PhysicsSimTest class>>appClass

^ PhysicsSim

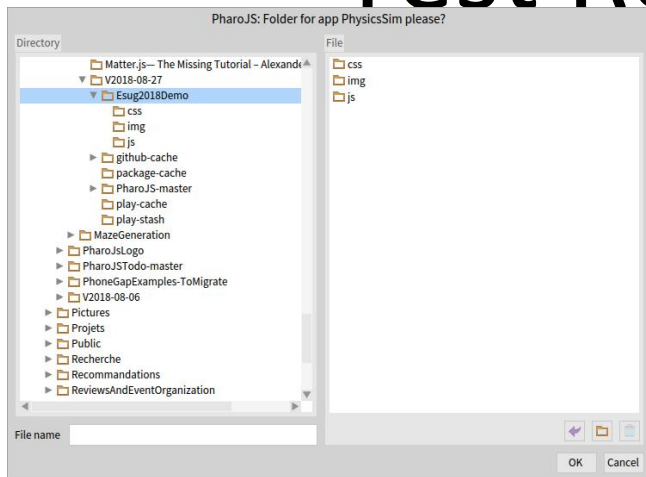


Entry point class

Test Requests App Folder



Test Requests App Folder



Folder where
HTML + other files
are located

A screenshot of a macOS Finder window showing the contents of the "Esug2018Demo" folder. The window title is "Esug2018Demo". The toolbar includes icons for view modes, settings, sharing, and search. The main area displays a list of files and folders with columns for Name, Date Modified, Size, and Kind.

Name	Date Modified	Size	Kind
▼ css	Today 17:51	--	Folder
index.css	Today 17:51	912 bytes	CSS
▼ img	18 Jun 2018 17:12	--	Folder
matterJsLogo.png	18 Jun 2018 17:12	15 KB	PNG image
pharoJsLogo.png	31 May 2018 16:39	31 KB	PNG image
index.html	Today 17:53	861 bytes	HTML
▼ js	Today 17:50	--	Folder
matter.js	21 Jun 2018 16:36	346 KB	JavaScript

HTML File = App's View

...

```
<button id="resetButton" class="button">Reset</button>  
<div id="simulationView"></div>
```

...

```
<script src="js/matter.js"></script>
```

```
<script src="index.js"></script>
```

...



Third-party
JS Library

HTML File = App's View

...

```
<button id="resetButton" class="simulationButton">
```

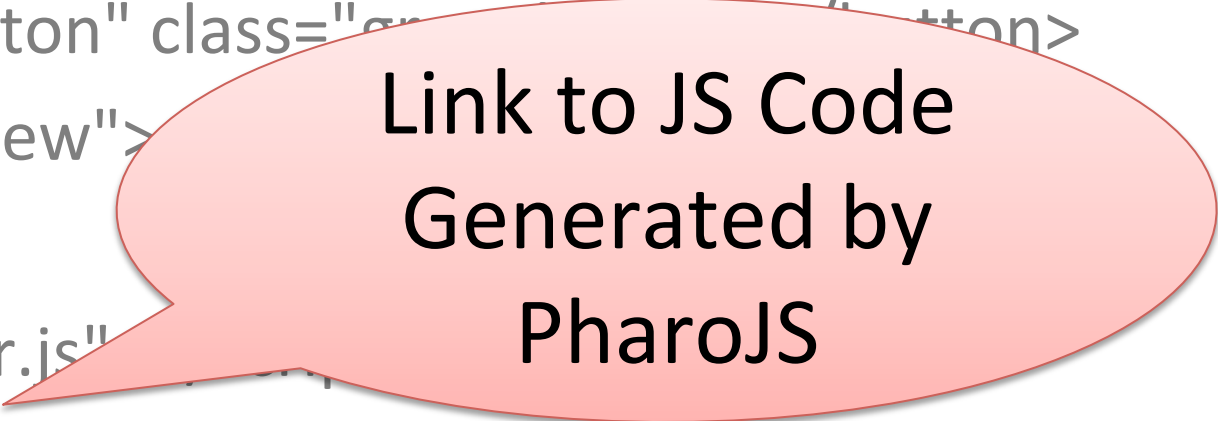
```
<div id="simulationView">
```

...

```
<script src="js/matter.js">
```

```
<script src="js/index.js"></script>
```

...



Link to JS Code
Generated by
PharoJS

HTML File = App's View



No Behavior!

...

```
<button id="resetButton" class="green">Reset</button>
```

```
<div id="simulationView"></div>
```

...

```
<script src="js/matter.js"></script>
```

```
<script src="js/index.js"></script>
```

...

HTML File = App's View

No Behavior!

...

```
<button id="resetButton" class="green">Reset</button>
```

```
<div id="simulationView"></div>
```

...

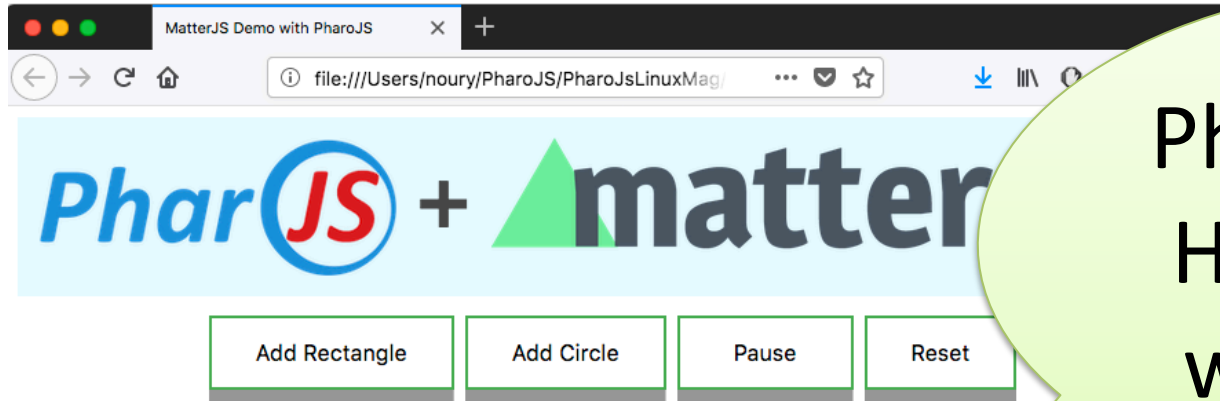
```
<script src="js/matter.js"></script>
```

```
<script src="js/index.js"></script>
```

...

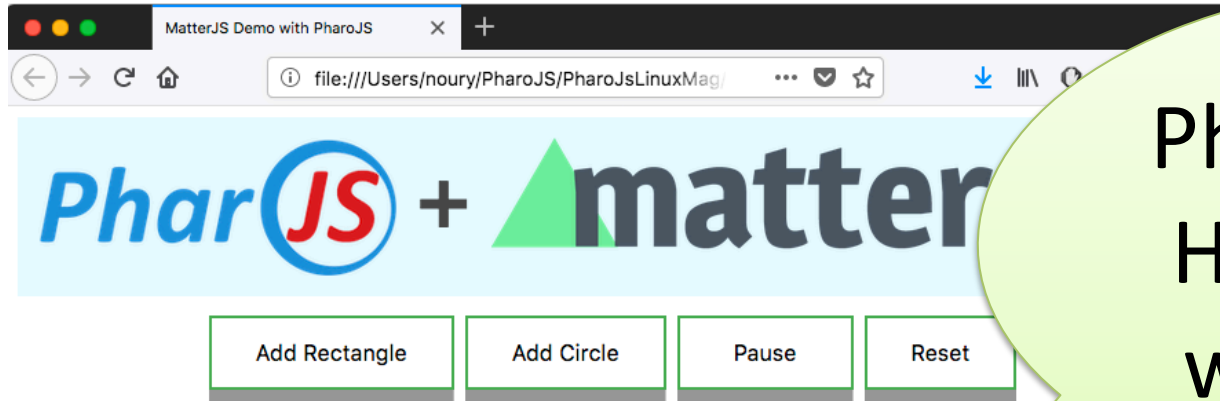
ID = to attach
Pharo code

Rerun Tests



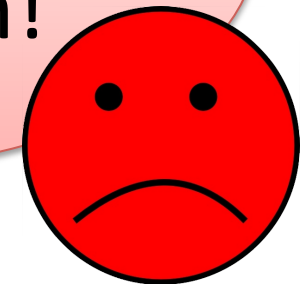
PharoJS Opens
HTML file in a
web browser

Rerun Tests

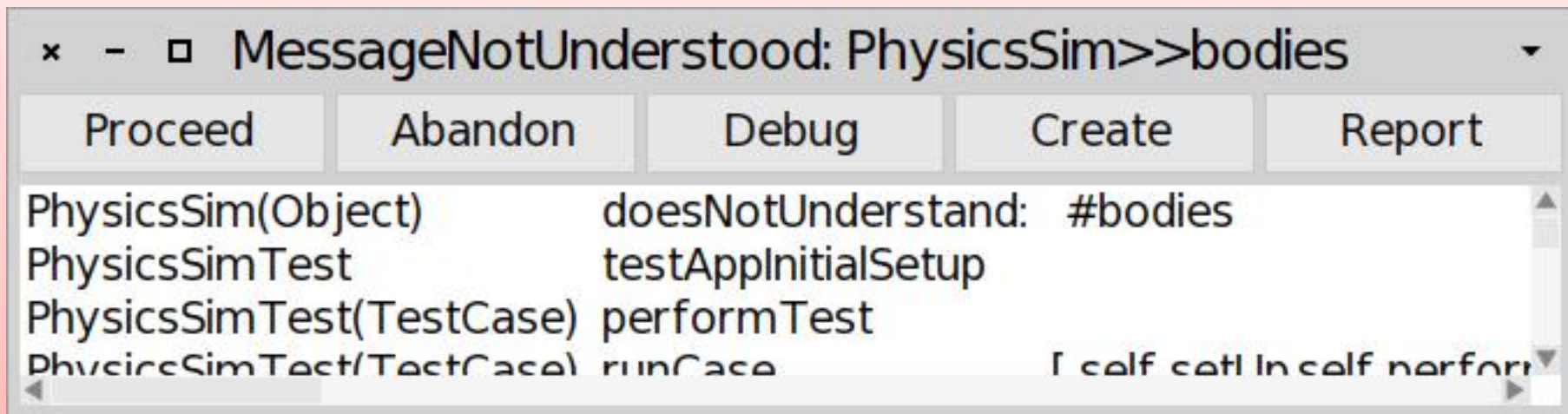


PharoJS Opens
HTML file in a
web browser

No
Simulation!



Rerun Tests



Fixing the App

JS Doc/Examples



Pharo Code

The screenshot shows the GitHub repository page for 'liabru/matter-js Wiki'. The page title is 'Home' and it includes a 'Jump to bottom' link. A note indicates 'Liam edited this page on Oct 18, 2017 · 16 revisions'. The 'matter.js' logo is prominently displayed, with the tagline 'Matter.js is a JavaScript 2D rigid body physics engine for the web'. Below the logo, there is a section titled 'Wiki pages' with a list of nine links: 1. Getting started, 2. Running, 3. Rendering, 4. Using plugins, 5. Creating plugins, 6. List of plugins and tools, 7. Tutorials, 8. Gallery, and 9. References.

PhysicsSim>>initialize

super initialize.

self createAndStartEngine.

self setupAndStartRendering.

PhysicsSim>>bodies

^ engine world bodies

Fixing the App

PhysicsSim>>matterJsRoot

^ **window** Matter

PhysicsSim.createAndStartEngine

| n

Proxy to
Javascript global

ine create.

ner create.

runner with: engine

Fixing the App

PhysicsSim>>matterJsRoot

^ window **Matter**

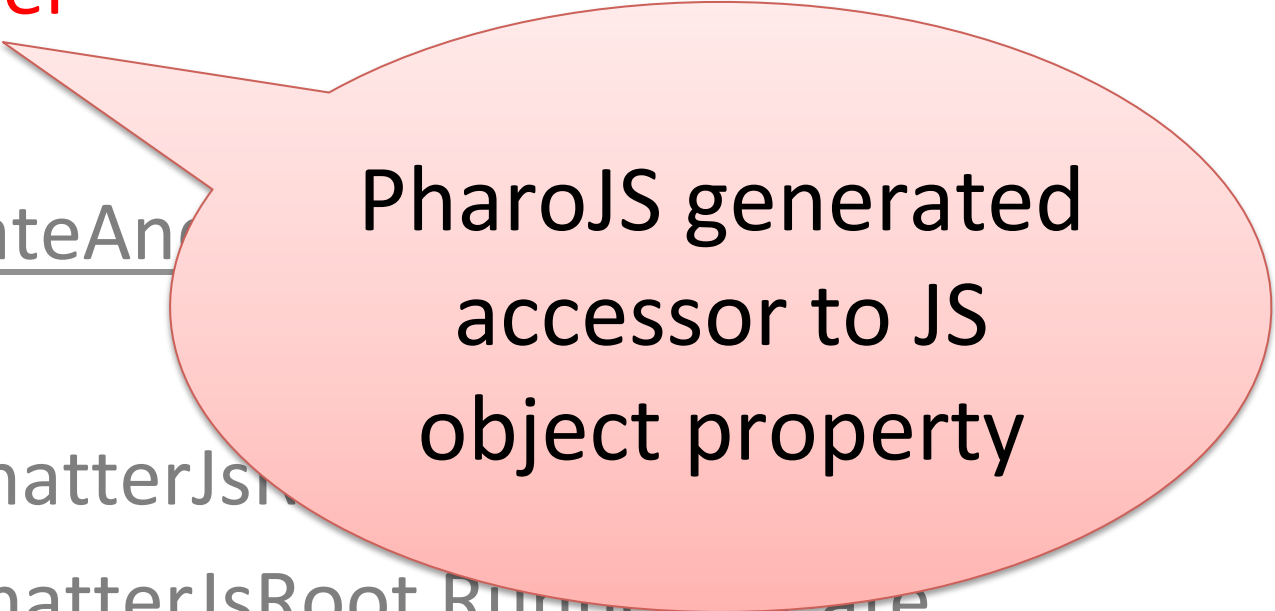
PhysicsSim>>createAn

| runner |

engine := self matterJsRoot

runner := self matterJsRoot Runner create.

self matterJsRoot Runner run: runner with: engine



PharoJS generated
accessor to JS
object property

Fixing the App

PhysicsSim>>matter

^ window Mat



Calling JS method
create()

PhysicsSim>>createAn

| runner |

engine := self matterJsRoot Engine **create**.

runner := self matterJsRoot Runner **create**.

self matterJsRoot Runner run: runner with: engine

Fixing the App

PhysicsSim>>matterJsRoot

^ window Matter

Phy

Calling JS method

run(*runner, engine*)

engine create.

runner := self matterJsRoot runner create.

self matterJsRoot Runner **run:** *runner with: engine*

Fixing the App

PhysicsSim>>setupAndStartRendering

| render simulationView |

simulationView := self elementAt: #simulationView.

render := self matterJsRoot Render create:

 {#element -> simulationView.

 #engine -> engine.

 #options -> {#width -> 800.

 #height -> 600.

 #wireframes -> false.

 #background -> 'transparent'}} asJsObject.

self matterJsRoot Render run: render

Finds the DOM element
with ID "SimulationView"

```
| render simulationView := self elementAt: #simulationView.  
render := self matterJsRoot Render create:
```

```
{#...  
#  
# ...  
<div id="simulationView"></div>  
...  
}
```

HTML

object.

```
self matterJsRoot Render run. render
```

Fixing it

Create a JS Object
from a Pharo
Dictionary

```
PhysicsSim>>se
```

```
| render simulation
```

```
simulationView := self element simulationView.
```

```
render := self matterJsRoot render create:
```

```
{#element -> simulationView.
```

```
#engine -> engine.
```

```
#options -> {#width -> 800.
```

```
#height -> 600.
```

```
#wireframes -> false.
```

```
#background -> 'transparent'}} asJsObject.
```

```
self matterJsRoot Render run: render
```

Defining Literal **JS** Objects in **Pharo**

JS Doc/Examples

Javascript code

```
{element : simulationView.  
engine :engine.  
options : {  
  width : 800.  
  height : 600.  
  wireframes : false.  
  background : 'transparent'}  
}
```



Pharo Code

Pharo equivalent

```
{#element -> simulationView.  
#engine -> engine.  
#options -> {  
  #width -> 800.  
  #height -> 600.  
  #wireframes -> false.  
  #background -> 'transparent'}  
} asJsObject.
```

Rerun Tests

The screenshot shows an IDE window titled "TestFailure: Got 0 instead of 3." with "Bytecode GT" in the top right corner. The "Stack" panel shows the following entries:

- PhysicsSimTest testAppInitialSetup
- PhysicsSimTest(TestCase) performTest
- PhysicsSimTest(TestCase) performTest

The "Source" panel shows the following code:

```
testAppInitialSetup  
self assert: app bodies size equals: 3
```

The "Diff (actual vs. exp...) Objects (actual vs. expected) Inspector" panel shows a comparison of values:

Actual	Expected
0	3

The "Inspector" panel also includes a "Pretty print" checkbox.

Fixing the App

PhysicsSim>>initialize

super initialize.

self createAndStartEngine.

self setupAndStartRendering.

self addInitialBodies

Creating MatterJS Bodies

JS Doc/Examples



Pharo Code

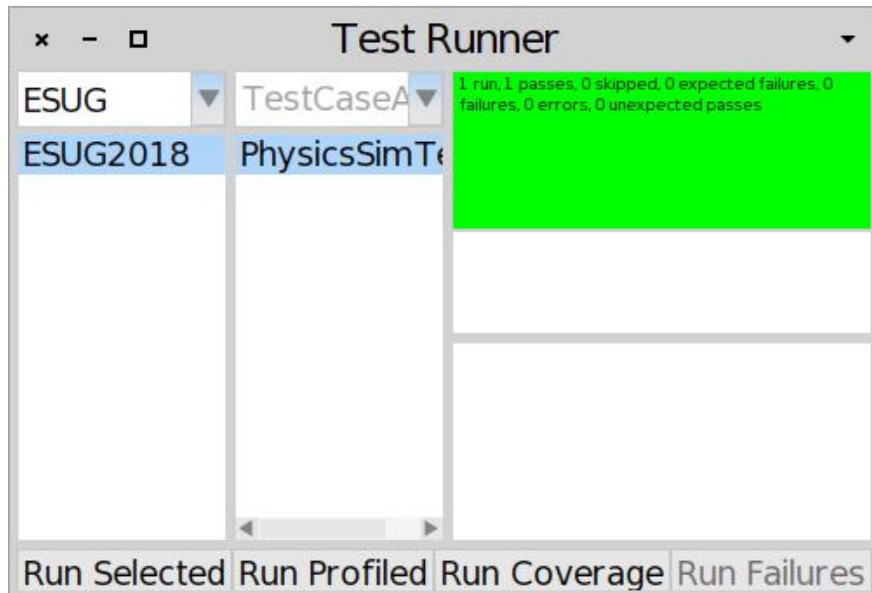
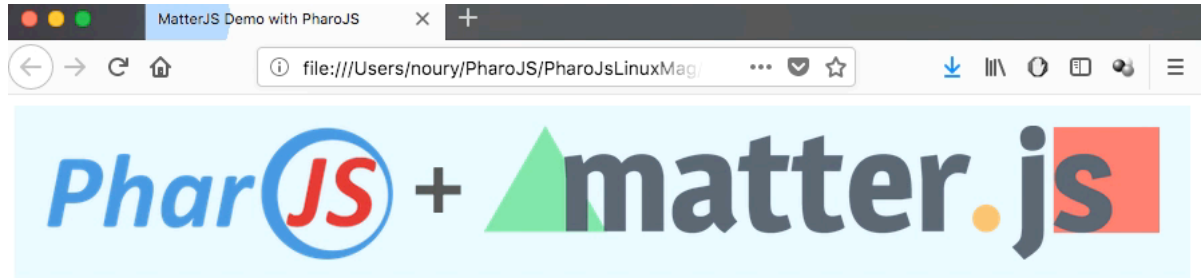
Javascript code

```
Bodies.rectangle(  
  400,  
  610,  
  810,  
  60,  
  { isStatic: true });
```

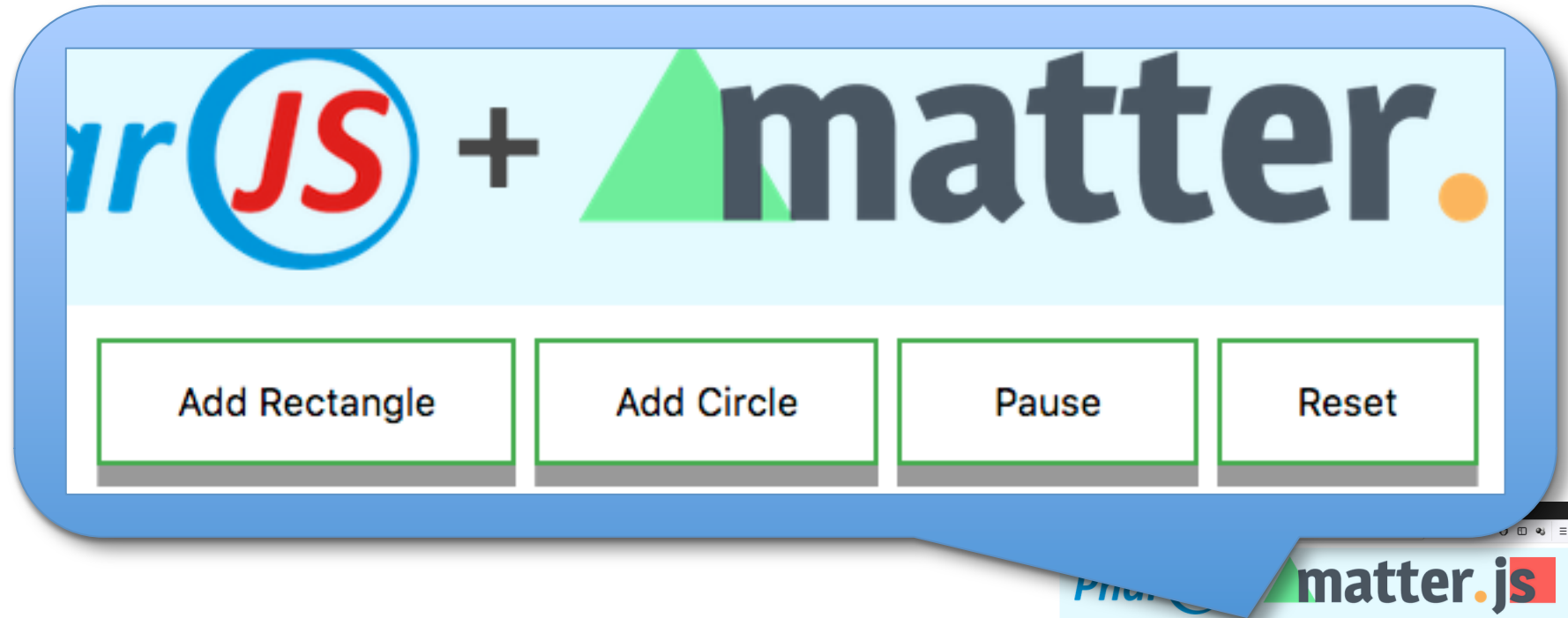
Pharo equivalent

```
self matterJsRoot Bodies  
  rectangle: 400  
  y: 610  
  width: 810  
  height: 60  
  options: {  
    #isStatic -> true} asJsObject.
```

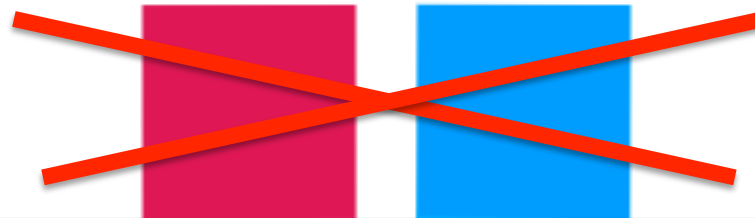
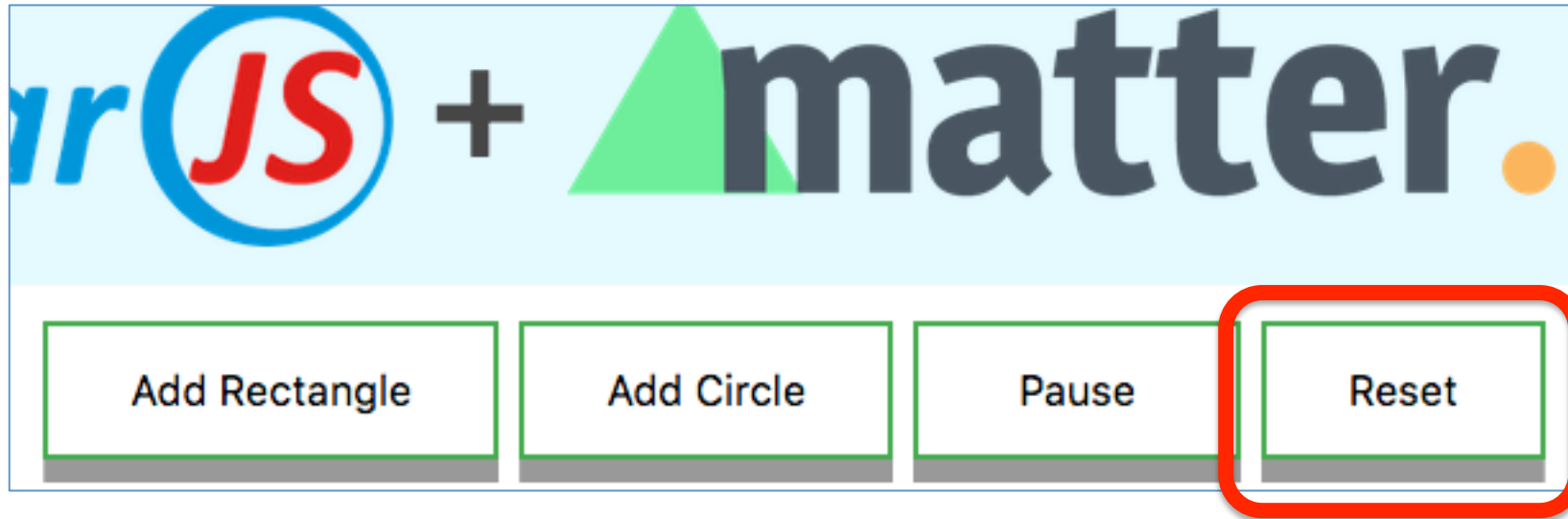
1st Test Passes!



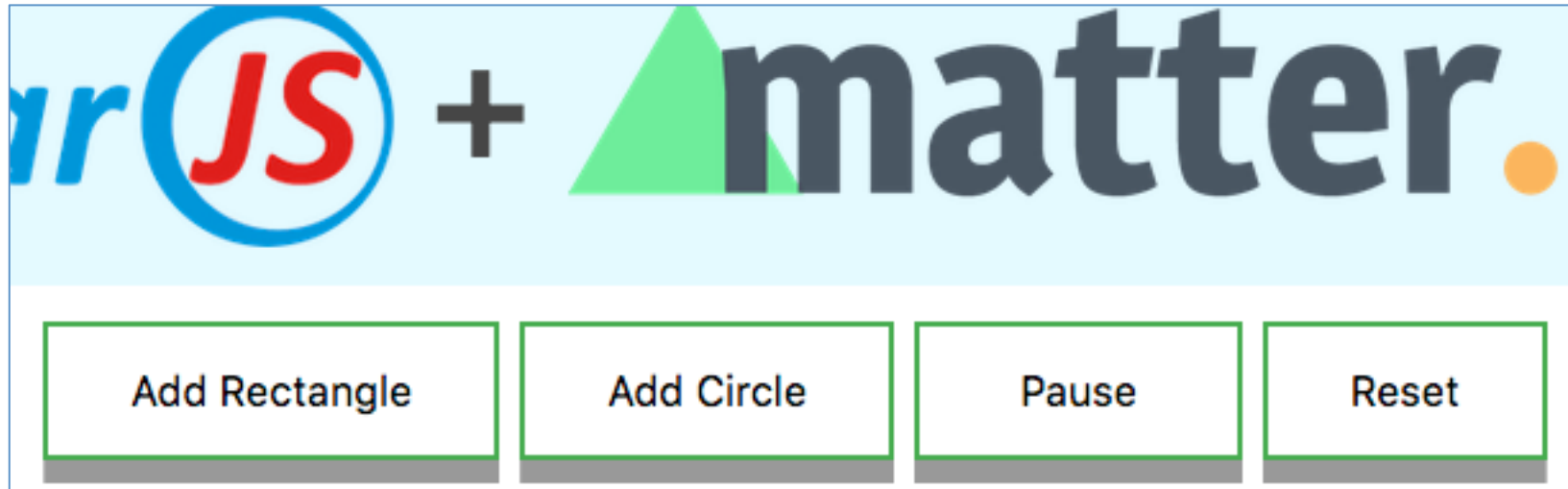
More features = Button Handling



Reset = Delete Mobile Bodies



HTML ID for Linking with Pharo



HTML

...

```
<button id="resetButton">Reset</button>
```

...

Write Tests for Reset Button

PhysicsSimTest>>testResetDeleteMobileBodiesOnly

self clickElementById: **#resetButton**.

self assert: **app bodies size** equals: **1**.

HTML

...

```
<button id="resetButton">Reset</button>
```

...

Write Tests for Reset Button

```
PhysicsSimTest>>testResetDeleteMobileBodiesOnly
```

```
self clickElementById: #resetButton.
```

```
self assert: ap ... bodies size equals: 1.
```

Pharo triggers event
on DOM object!

...

```
<button id="resetButton">Reset</button>
```

...

Test Fails

The image shows a test runner interface with a failure message: "TestFailure: Got 3 instead of 1." The stack trace includes:

- PhysicsSimTest testResetDeleteMobile
- PhysicsSimTest(TestCase) performTest

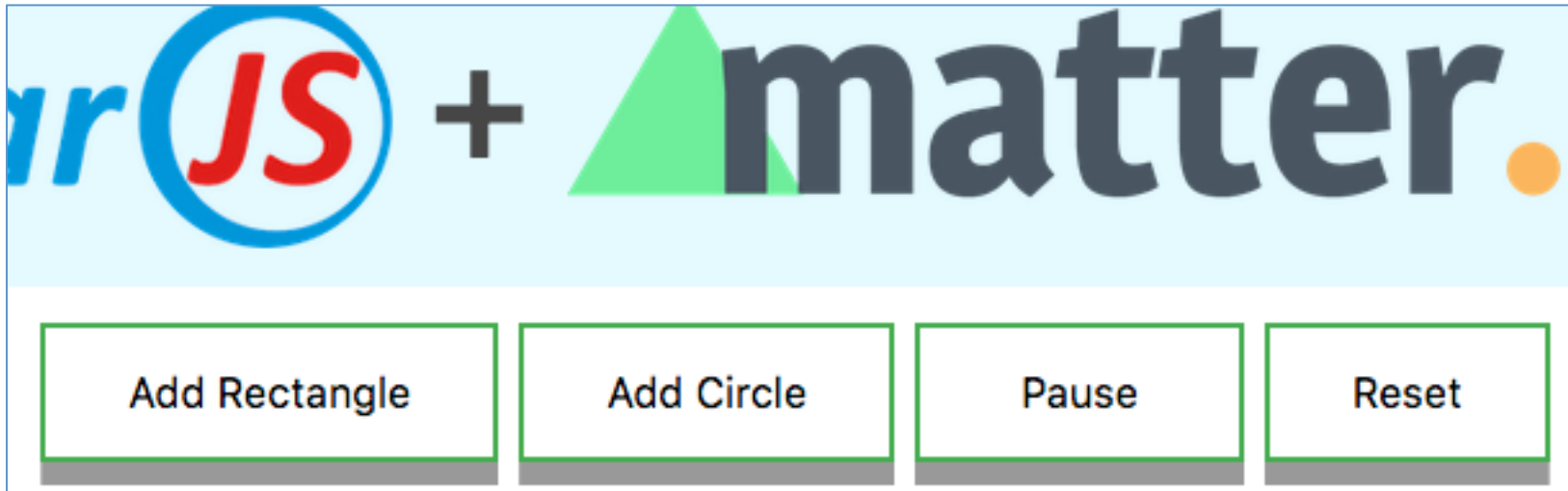
The source code for `testResetDeleteMobileBodiesOnly` is shown:

```
self clickElementById: #resetButton.  
self assert: app bodies size equals: 1.
```

The diff view shows the actual value is 3 and the expected value is 1.

Overlaid on the right is a browser window titled "MatterJS Demo with PharoJS" showing the "PharoJS + matter.js" logo and buttons for "Add Rectangle", "Add Circle", "Pause", and "Reset". The demo area contains two rectangles, one red and one blue.

Add Behavior to HTML Button



PhysicsSim>>initialize

...

self onClick: #resetButton

do: [self matterJsRoot World

clear: engine world

keepStatic: true]

Add Behavior to HTML Button



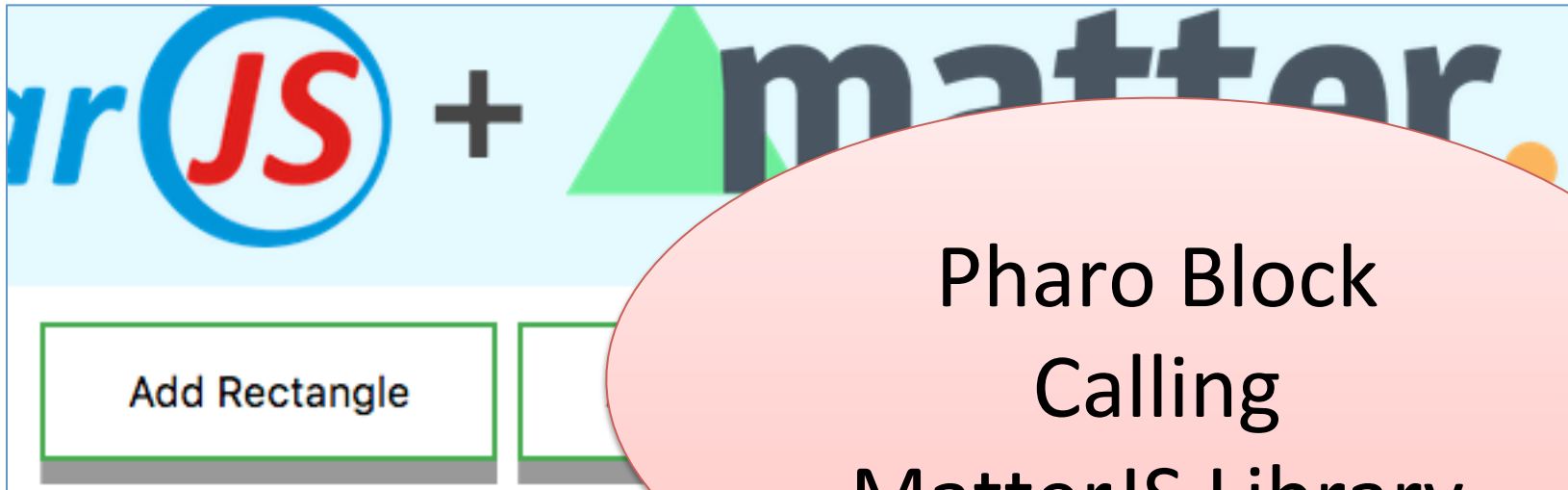
PhysicsSim>>initialize

...

self onClick: **#resetButton**

```
do: [ self matterJsRoot World  
      clear: engine world  
      keepStatic: true]
```


Add Behavior to HTML Button



Pharo Block
Calling
MatterJS Library

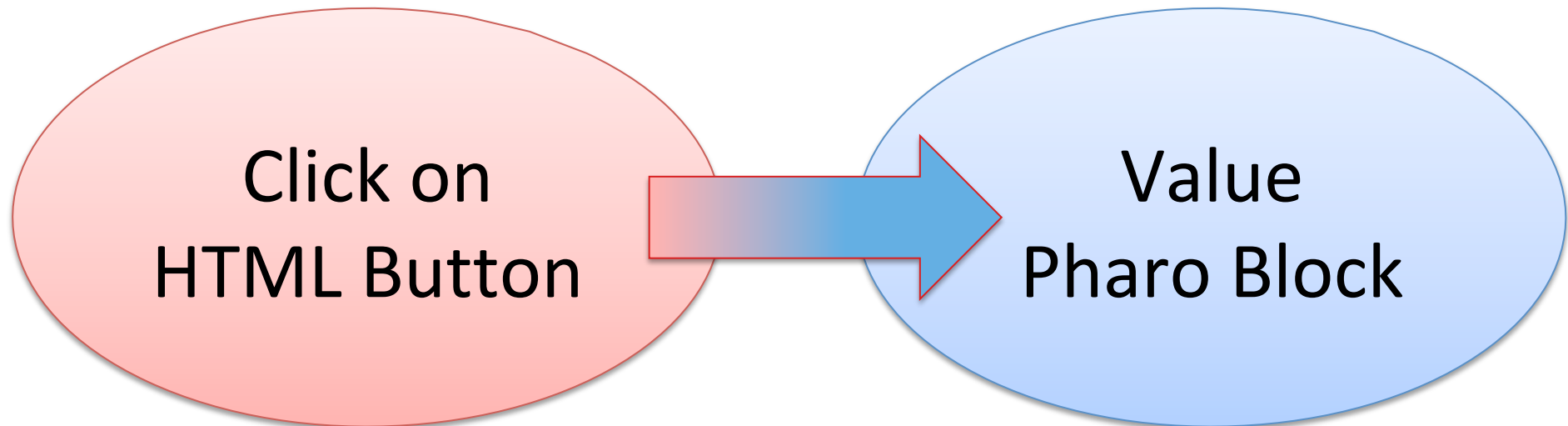
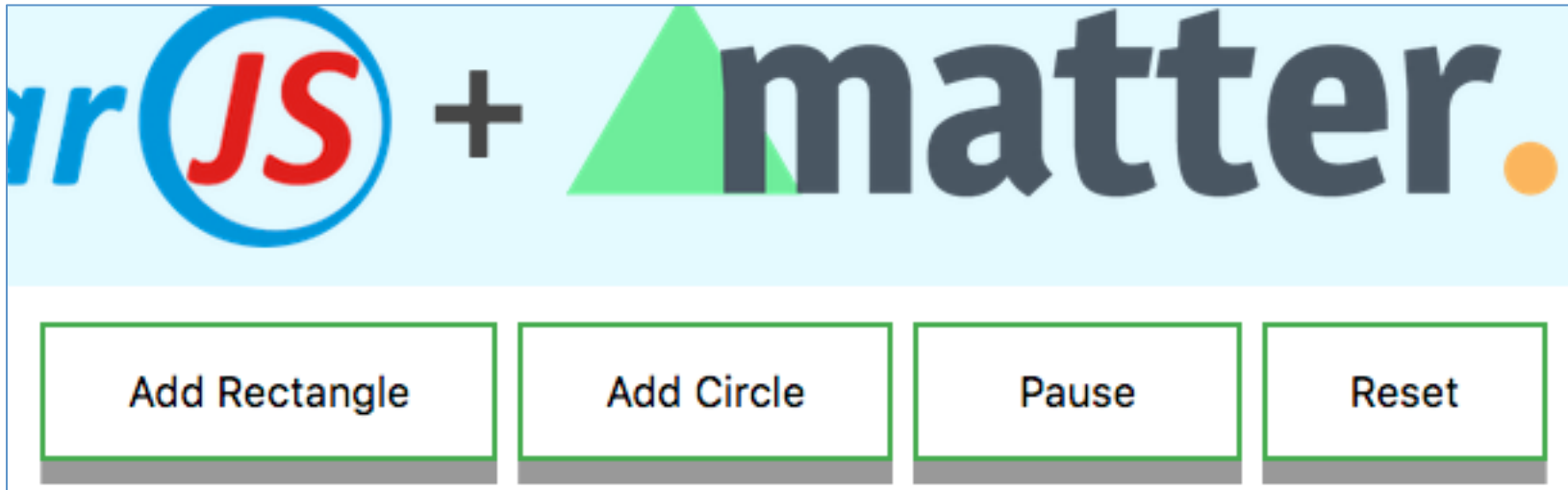
PhysicsSim>>initialize

...

self onClick: #resetButton

do: [self matterJsRoot World
clear: engine world
keepStatic: true]

Add Behavior to HTML Button



Lifecycle with *Phar* JS

100%

Pharo

0%



1. Write Tests



2. Pass the tests

3. Export to JS

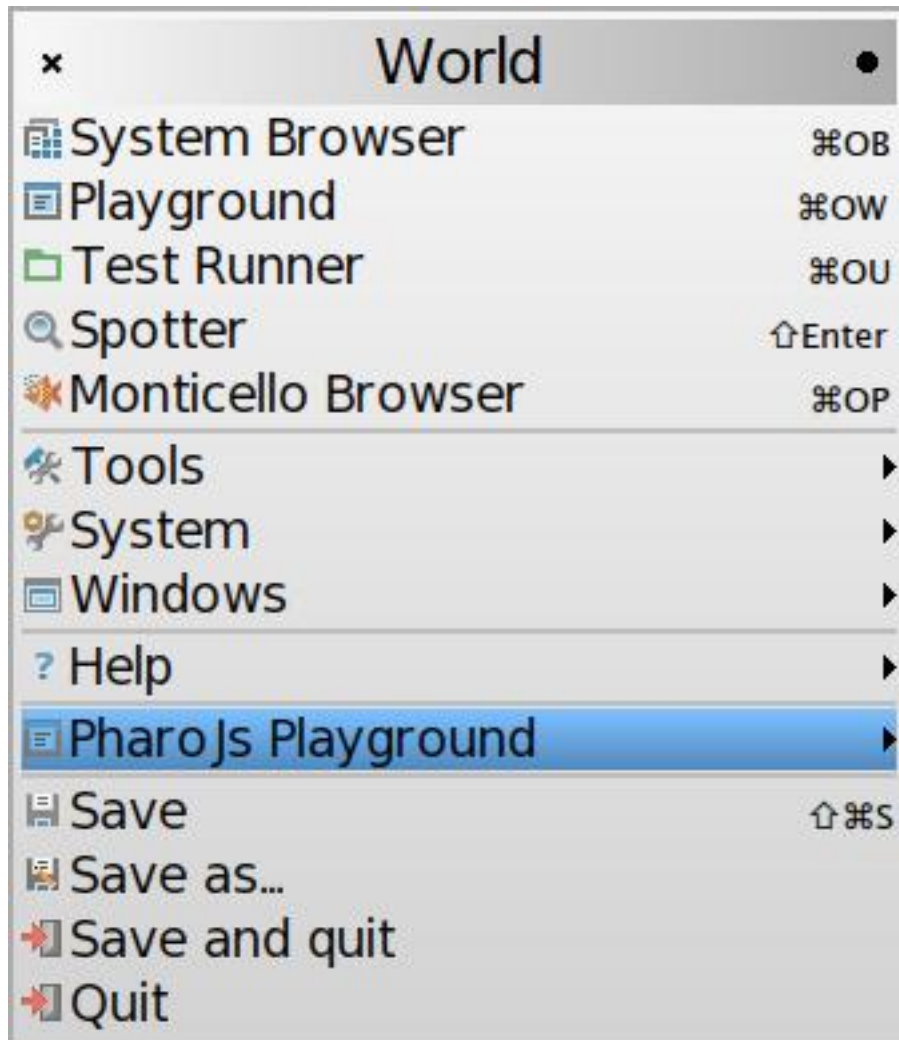
0%

Javascript

100%

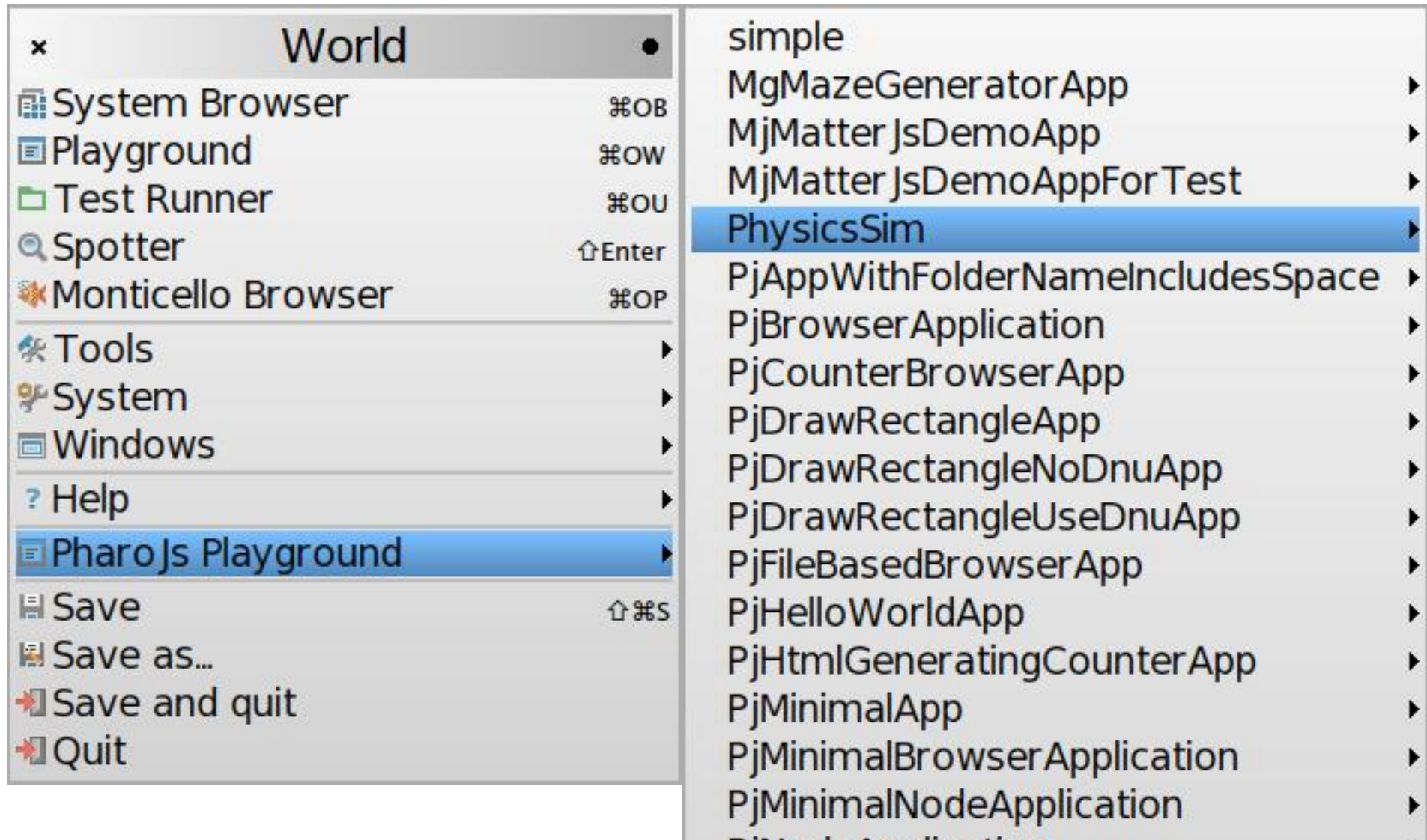
Export App for Production

100% Javascript



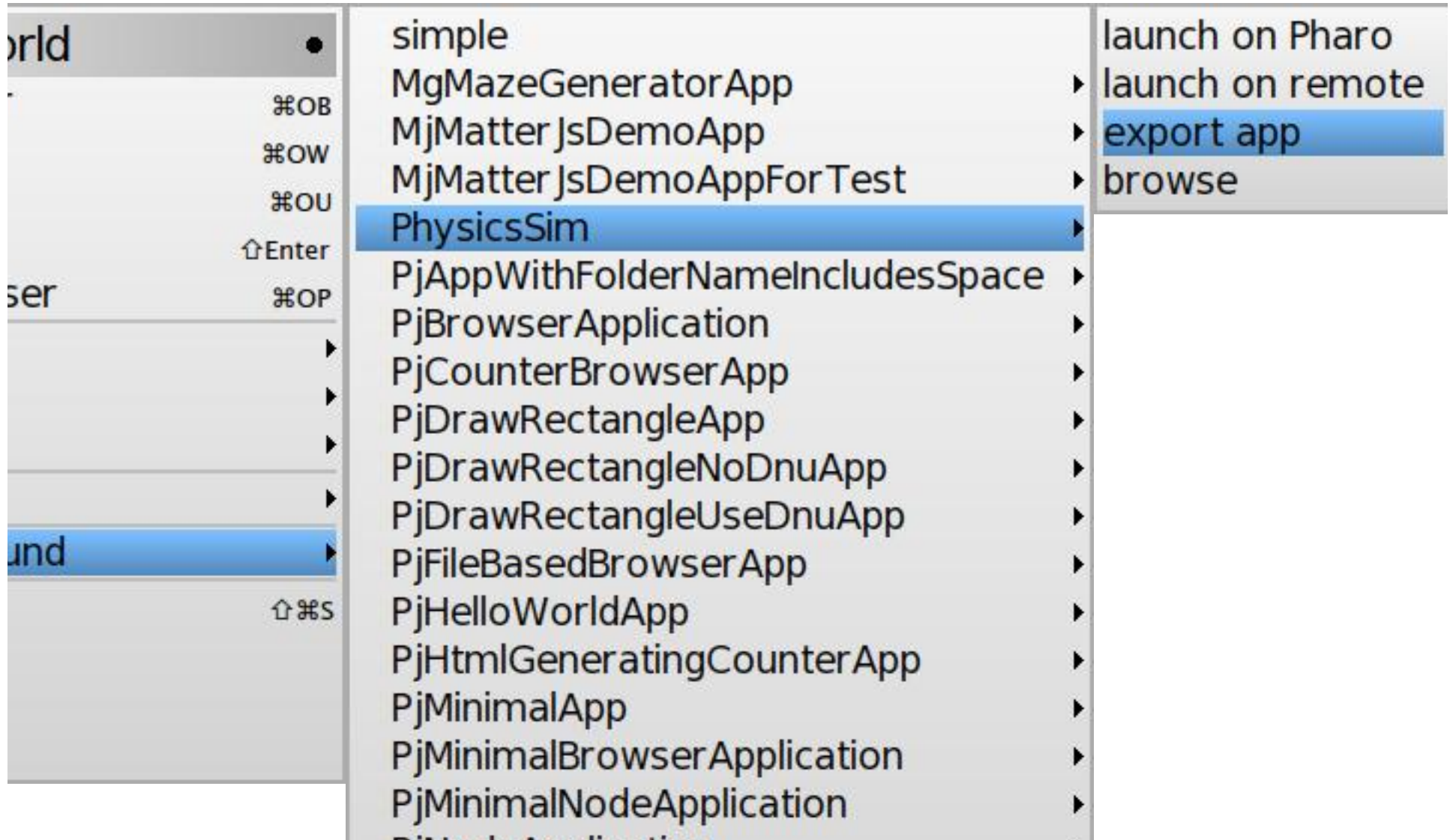
Export App for Production

100% Javascript



Export App for Production

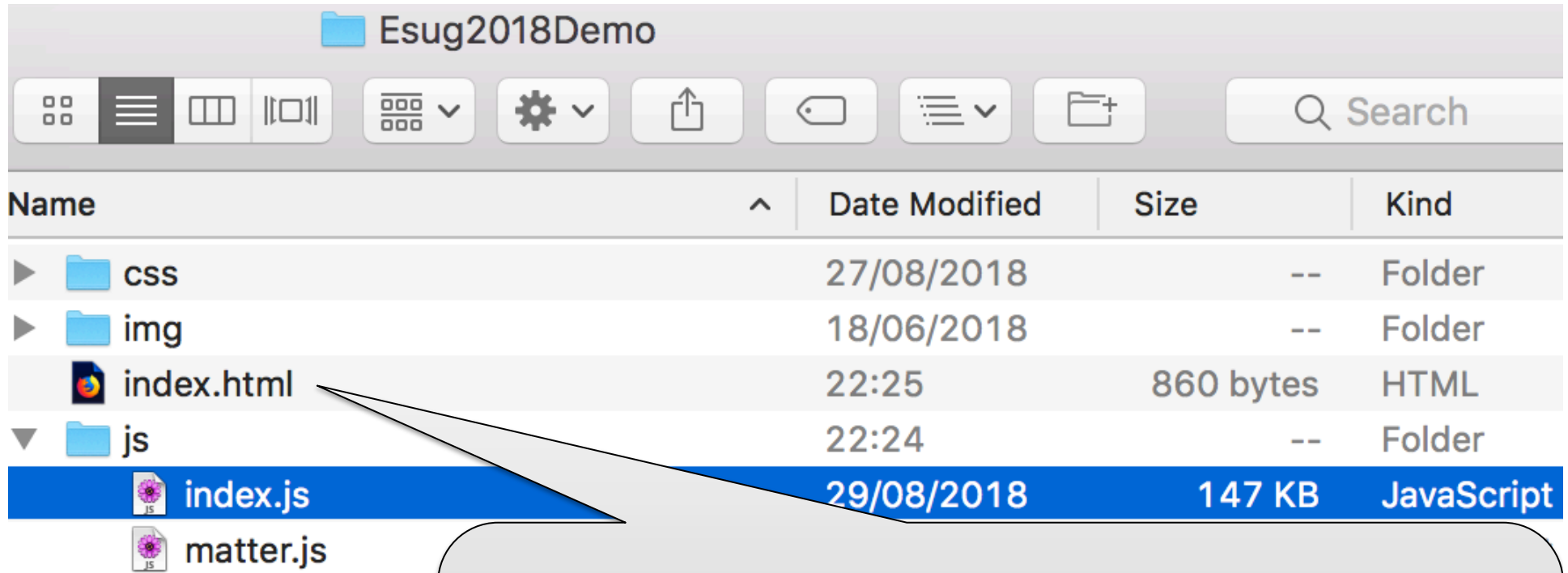
100% Javascript



Exported JS file

Name	^	Date Modified	Size	Kind
▶ css		27/08/2018	--	Folder
▶ img		18/06/2018	--	Folder
index.html		22:25	860 bytes	HTML
▼ js		22:24	--	Folder
index.js		29/08/2018	147 KB	JavaScript
matter.js		21/06/2018	346 KB	JavaScript

Exported JS file



Name	Date Modified	Size	Kind
css	27/08/2018	--	Folder
img	18/06/2018	--	Folder
index.html	22:25	860 bytes	HTML
js	22:24	--	Folder
index.js	29/08/2018	147 KB	JavaScript
matter.js			

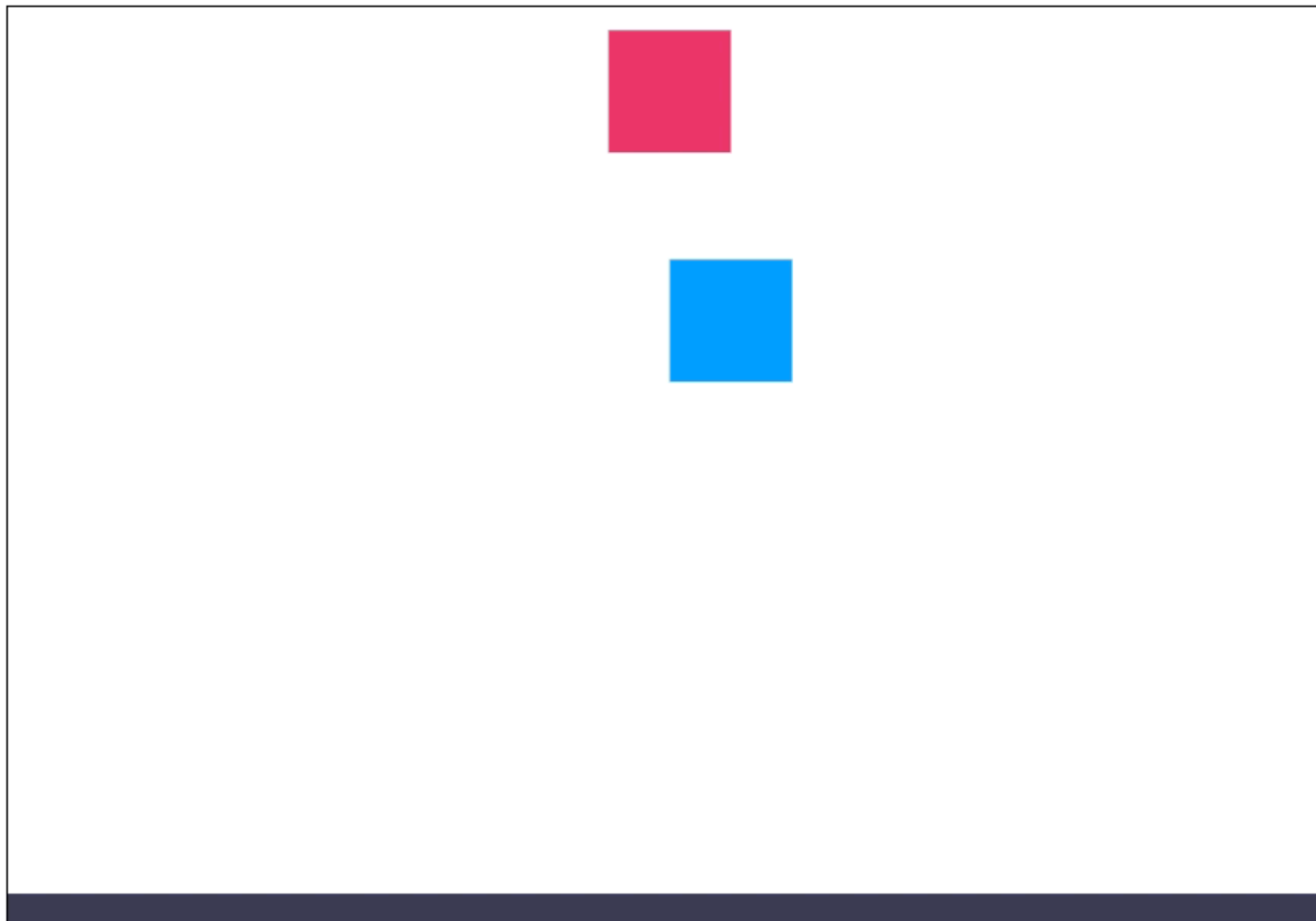
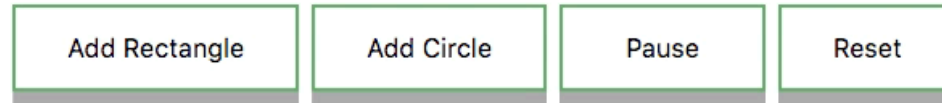
...

```
<script src="js/matter.js"></script>
```

```
<script src="js/index.js"></script>
```

...

<http://pharojs.org/Demo/MatterJsDemo/index.html>





Beyond Research!



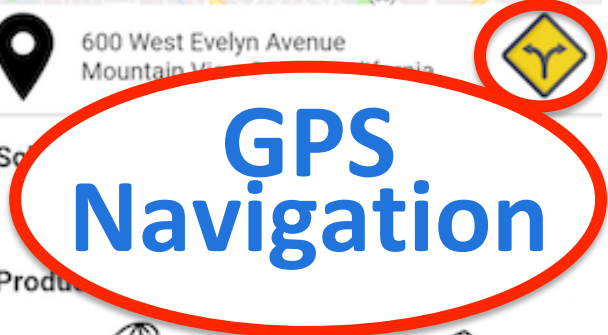
Open Source for the Real World

Pharo Tradition!

nootrix.com



nootrix.com



What Next?

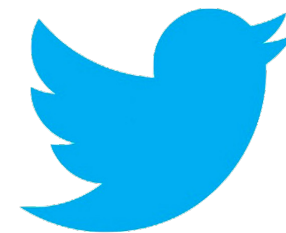
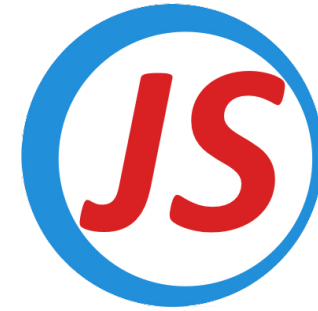
- Better support for mobile apps
- Migrate to Pharo 7
- Improve Code Extraction
- Middleware for hybrid apps

This
afternoon!

seaside 

Learn more about PharoJS

- Web: <http://pharojs.org>
 - FAQ + ...
 - Thanks ESUG for the support
- Slack: <https://pharojs.slack.com/>
 - Discussions
- Twitter: <https://twitter.com/pharojs>
 - News
 - Subscription to PharoJS Slack





Develop in Pharo
Run on Javascript

<http://pharojs.org>

Noury Bouraqadi & Dave Mason