Tutorial

Pharo-Based TDD for Javascript Apps

Noury Bouraqadi & Dave Mason

http://pharojs.org

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

Ryerson University

=

Develop in Pharo

+

Run on Javascript

PharJS

MIT License

=

Framework

+

Tools

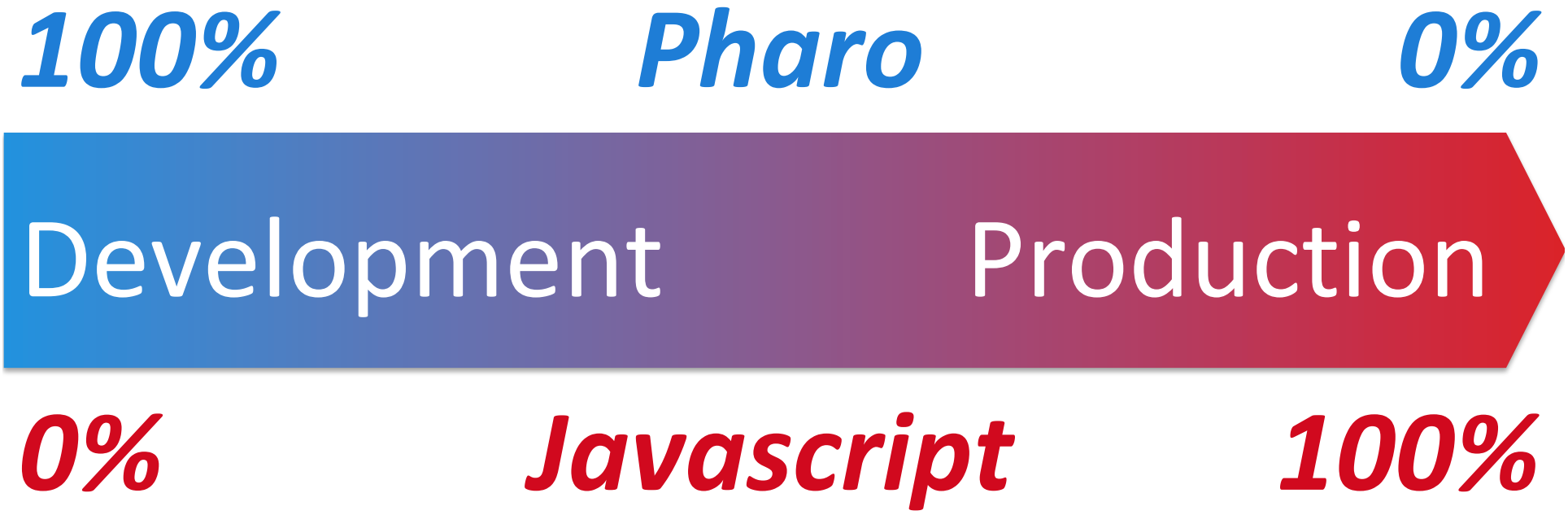
Why PharJS ?

- Develop in Smalltalk all the time ☺
- Portability

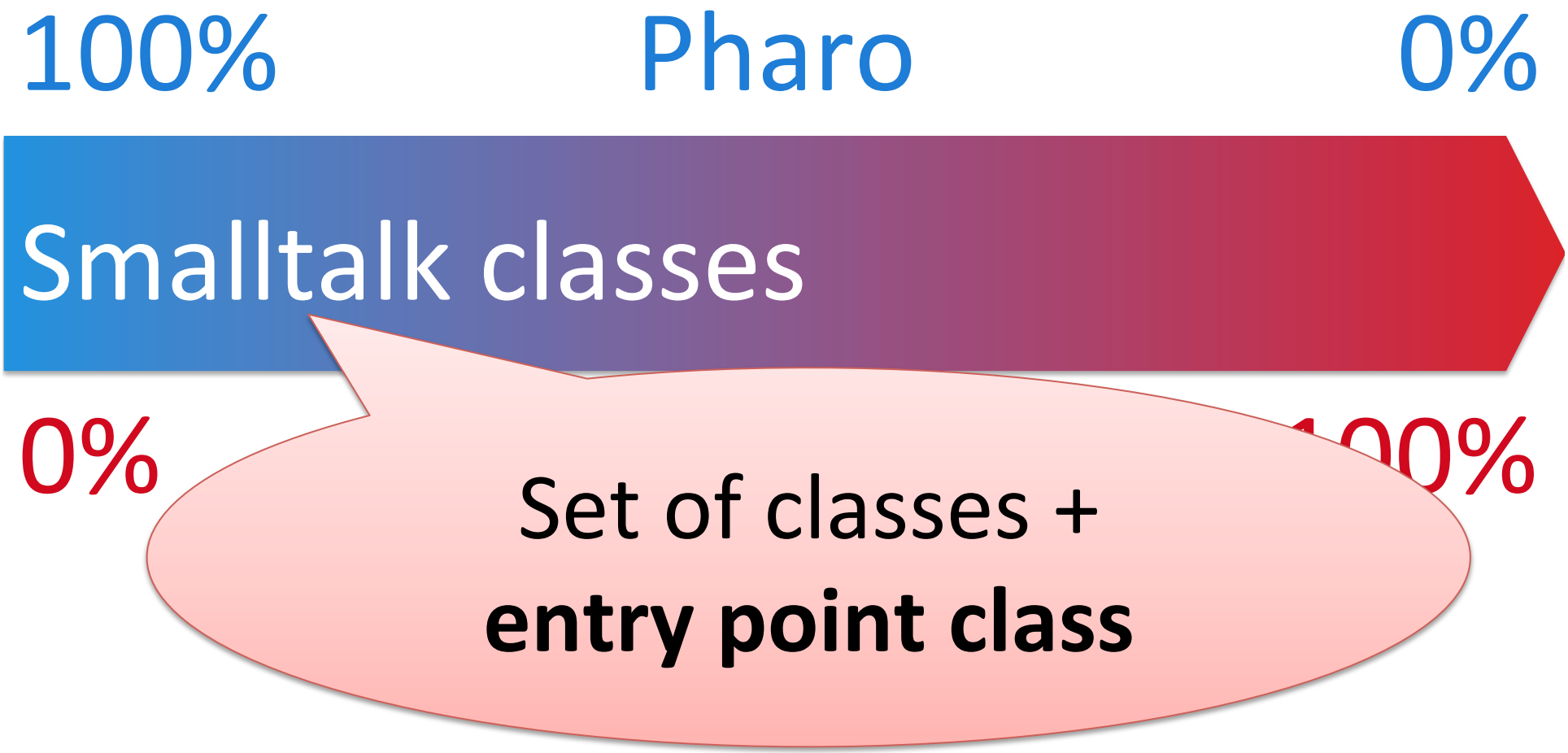
- Front-end/standalone

- Reuse existing JS libraries
- Run-time Speed

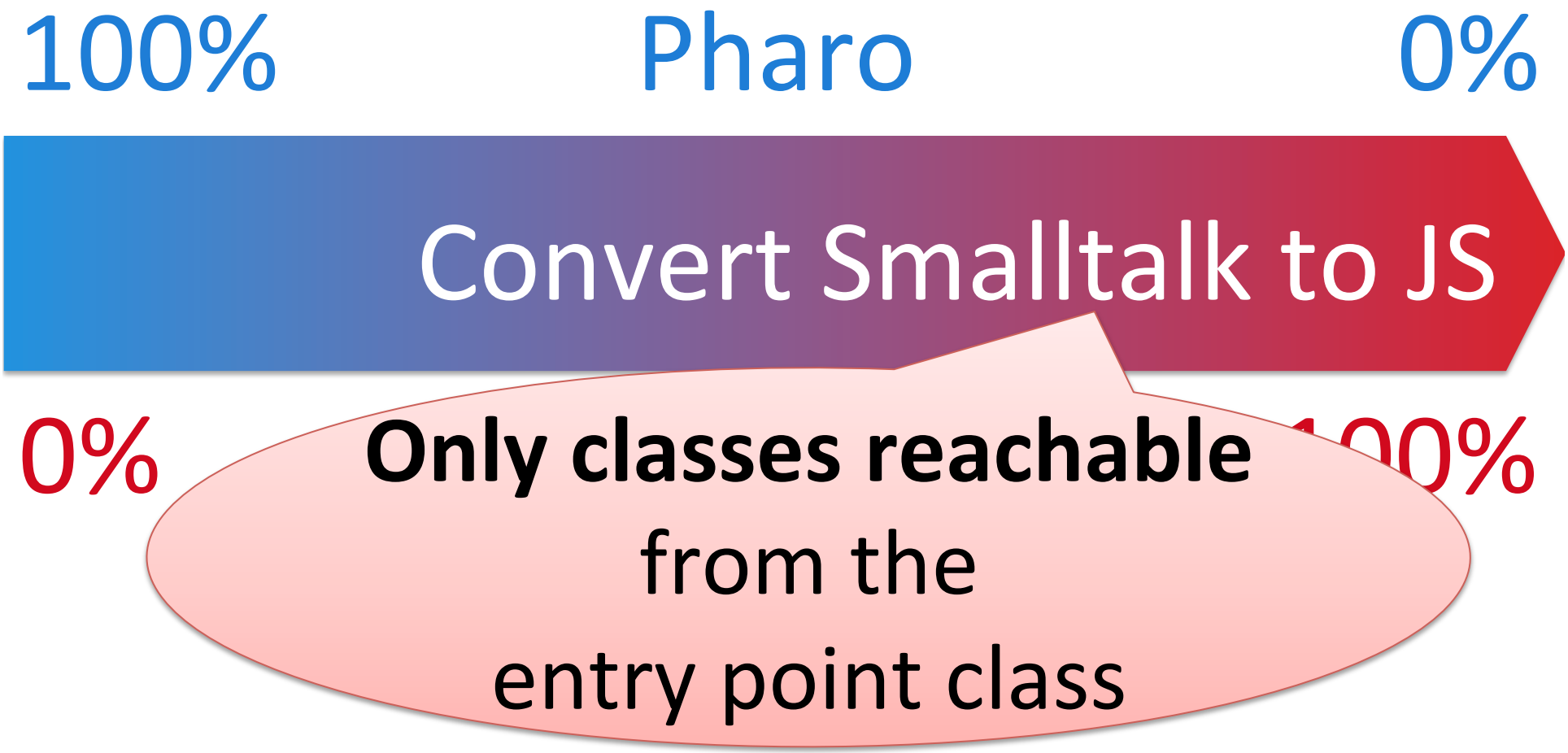
Develop in Pharo, Run on Javascript



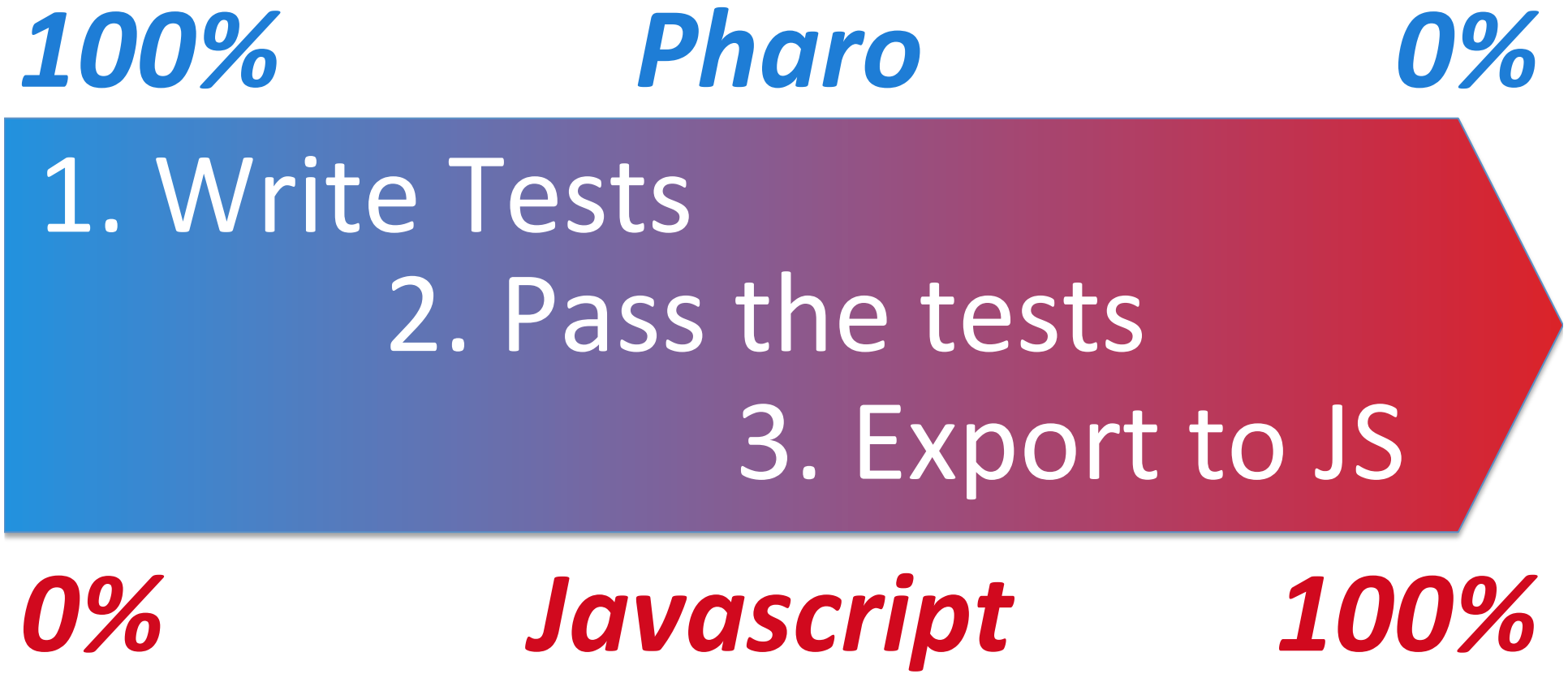
Develop in Pharo, Run on Javascript



Develop in Pharo, Run on Javascript

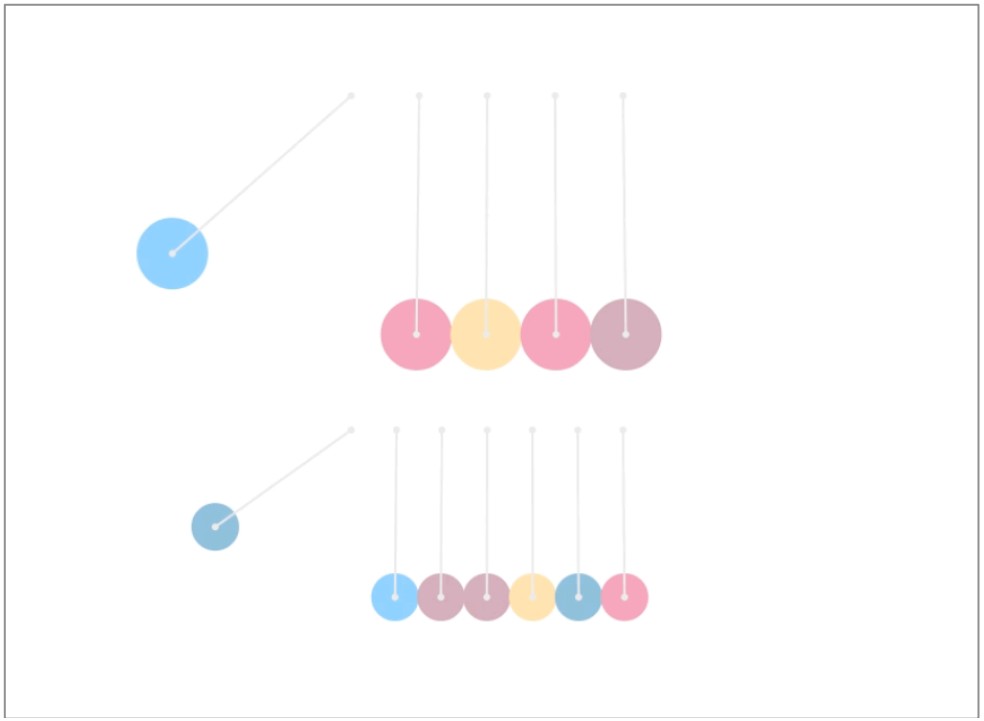


Lifecycle with PharJS



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

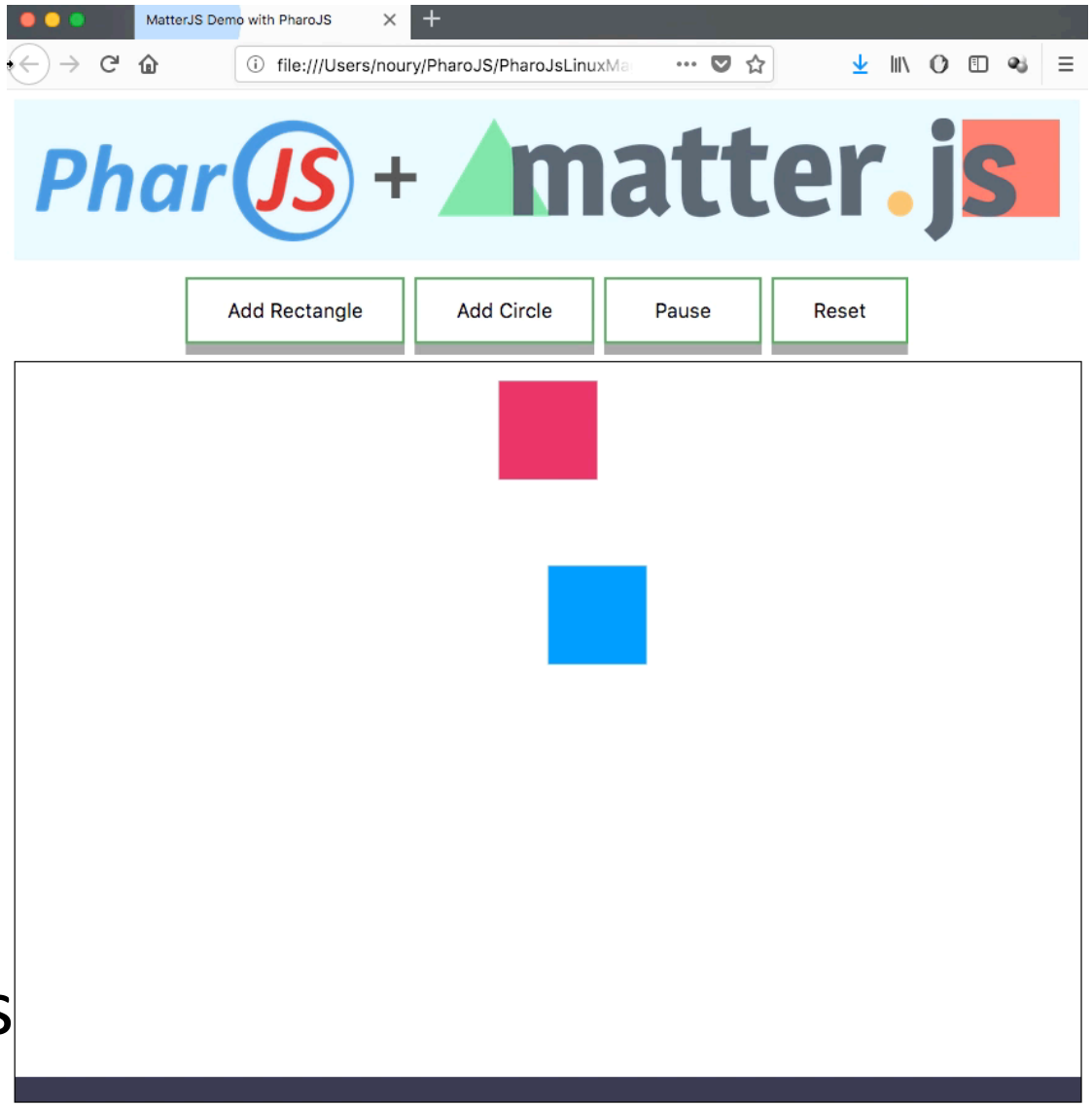
➔2 falling rectangles

- "Add xxx" buttons

➔ Adds a body

- "Reset" button

➔ Removes all bodies



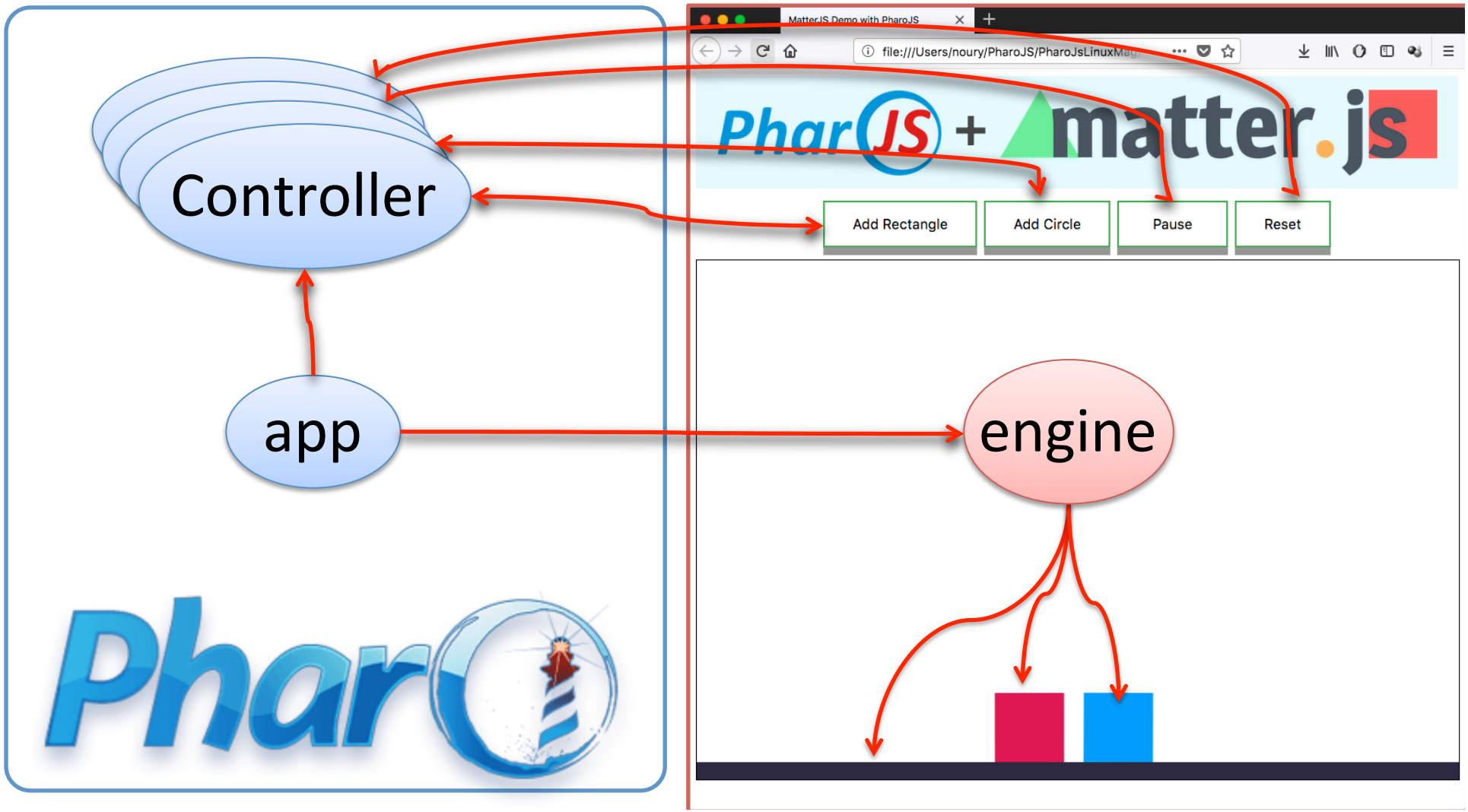
Write Tests 100% Pharo & 0% JS

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

Write Tests 100% Pharo & 0% JS


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

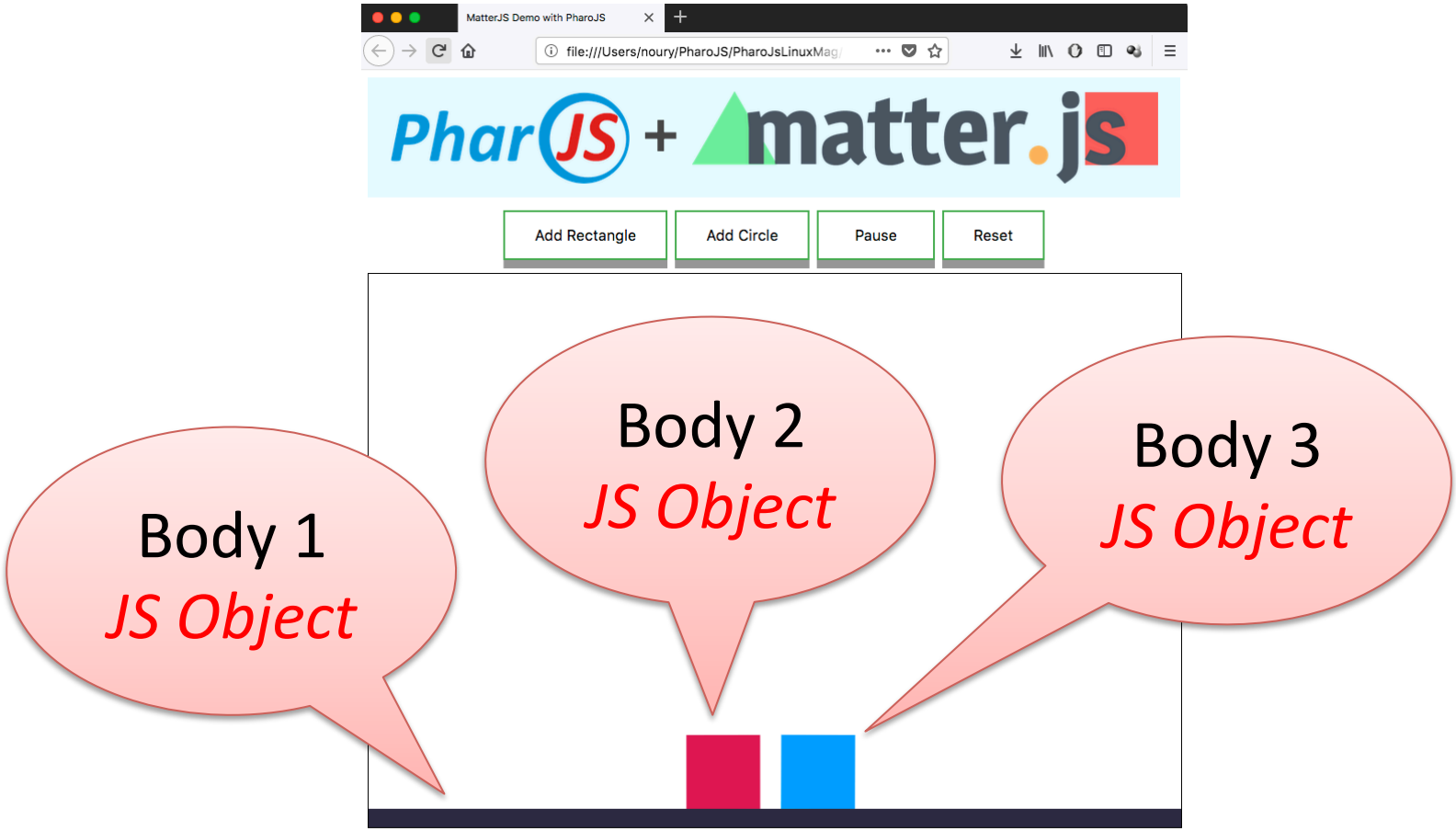
Running the Tests ~50% Pharo & ~50% JS



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

PhysicsSimTest>>testAppInitialSetup

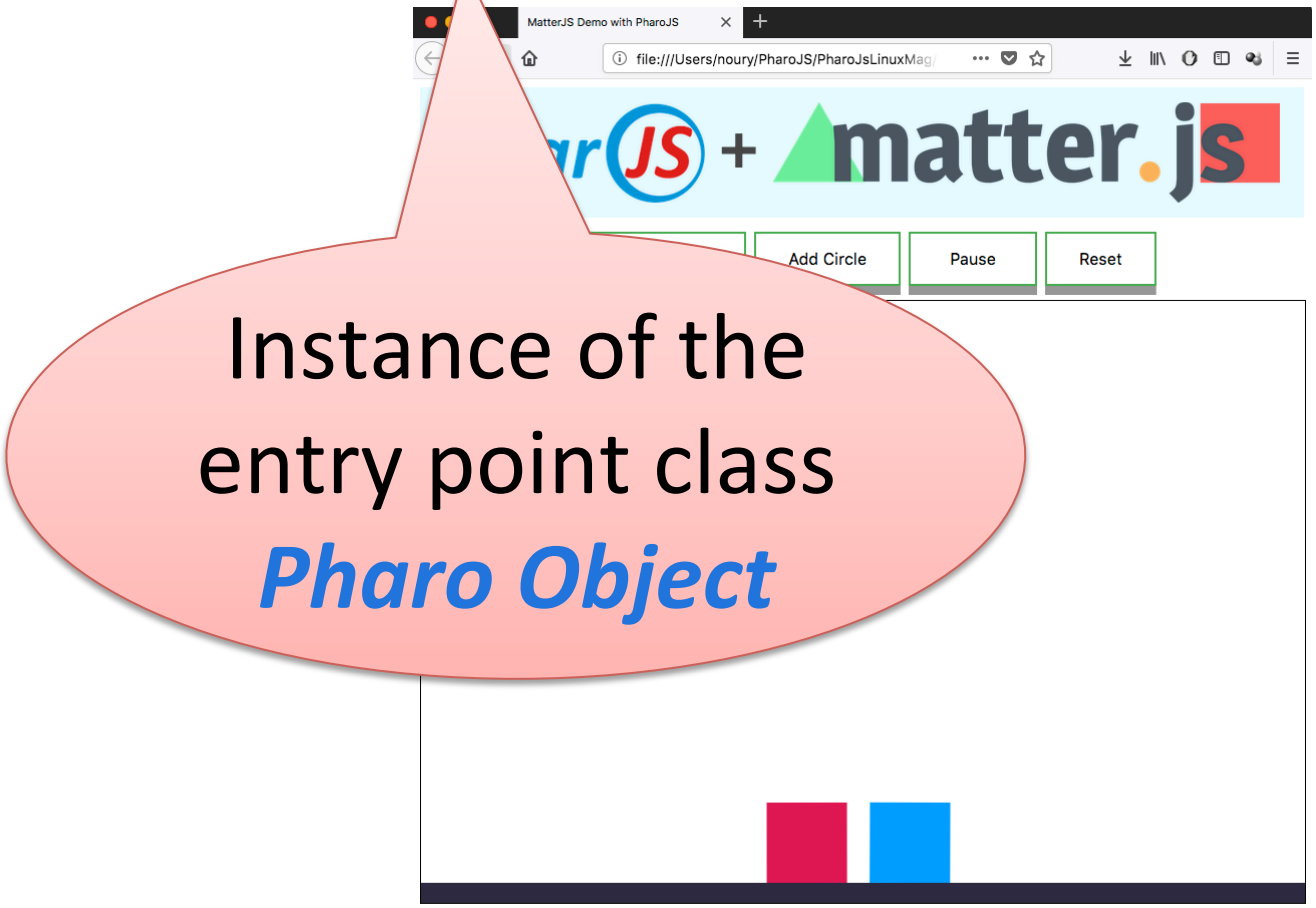
self assert: **app bodies size** equals: **3**.



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

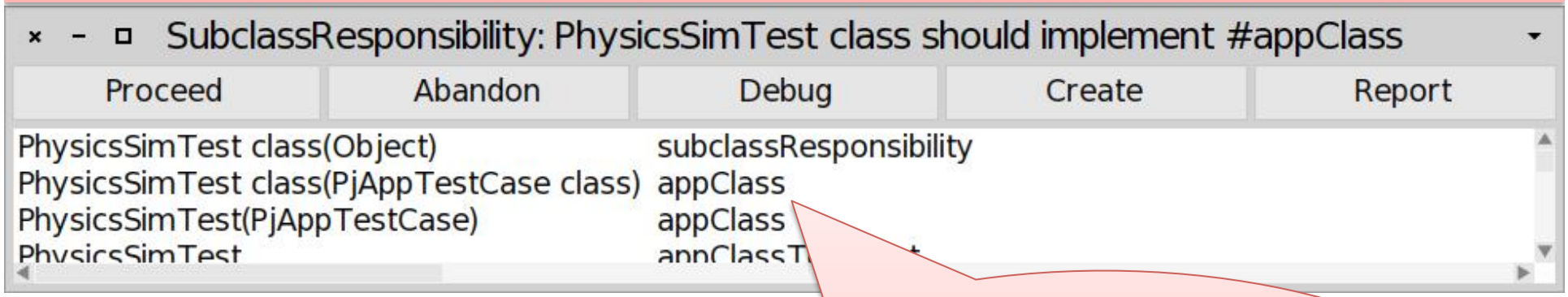
PhysicsSimTest>>testAppInitialSetup

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



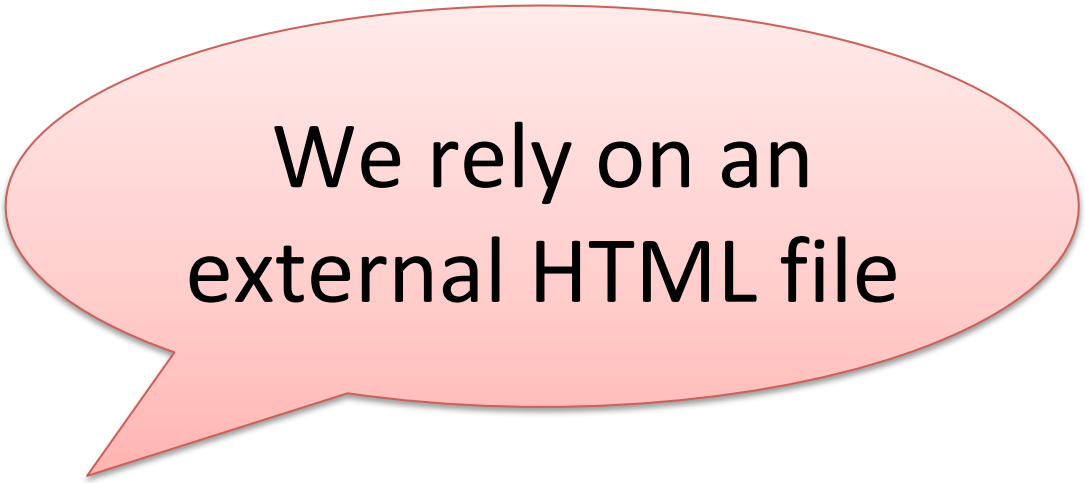
Running the Tests ~50% Pharo & ~50% JS

appClass subclassResponsibility



Missing
entry point class

Defining the Entry Point Class



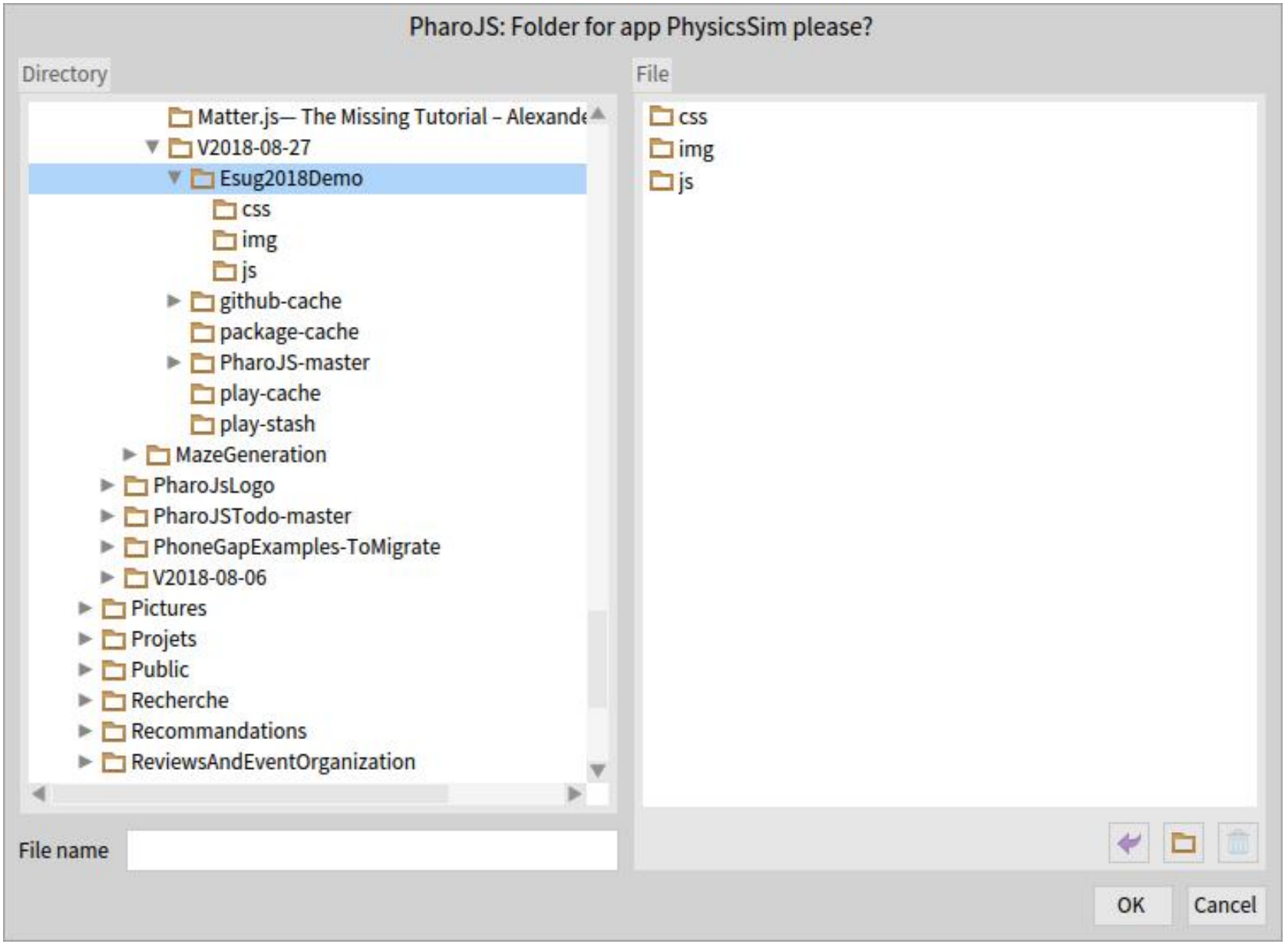

```
PjFileBasedWebApp subclass: #PhysicsSim
    instanceVariableNames: 'engine'
    classVariableNames: ''
    package: 'ESUG2018'
```

Test References Entry Point Class

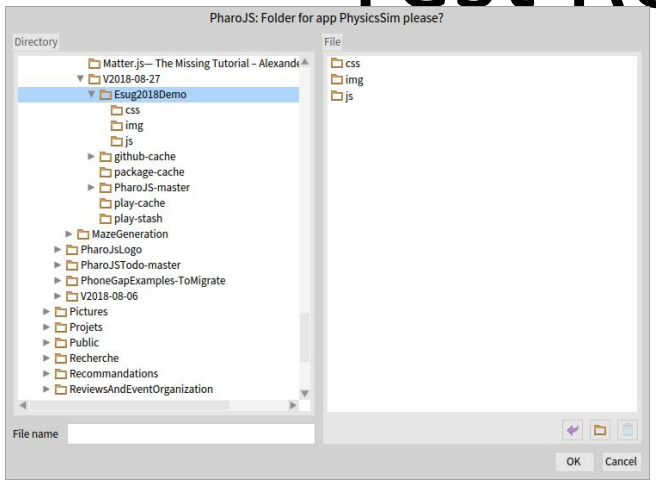
PhysicsSimTest class>>appClass

^ PhysicsSim

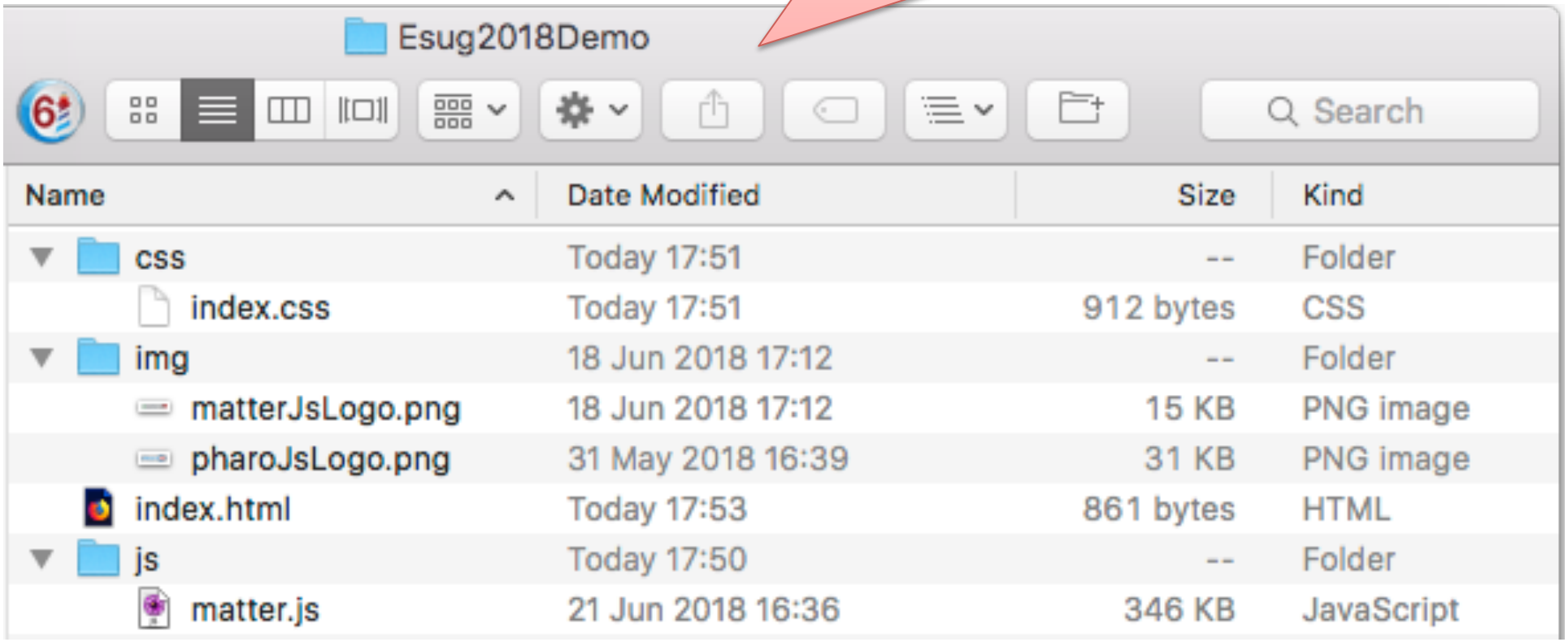
Test Requests App Folder



Test Requests App Folder

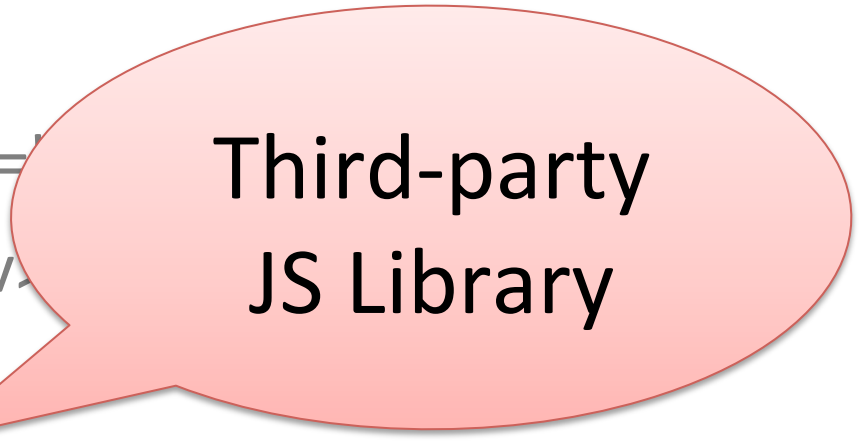


Folder where HTML + other files are located



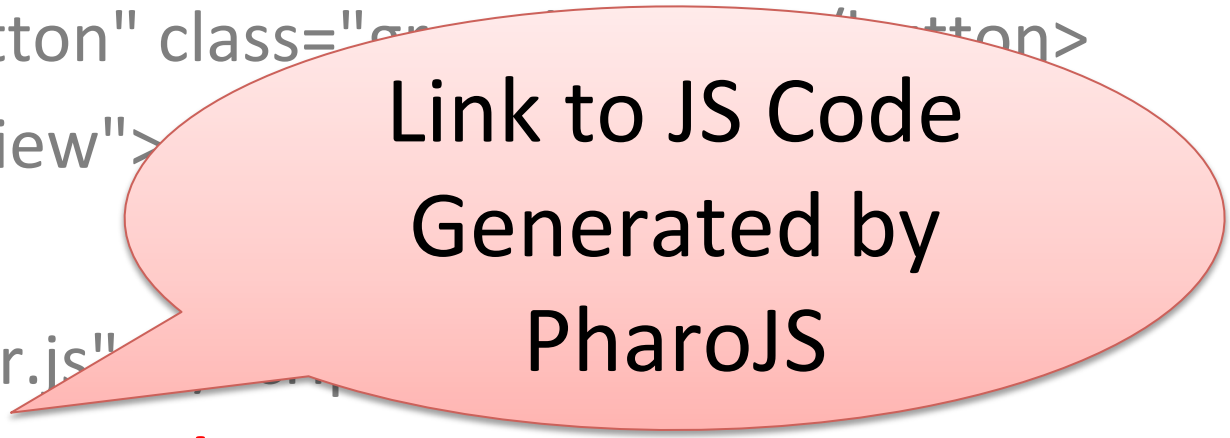
HTML File = App's View

```
...
<button id="resetButton" class=
<div id="simulationView"></div>
...
<script src="js/matter.js"></script>
<script src="index.js"></script>
...
```

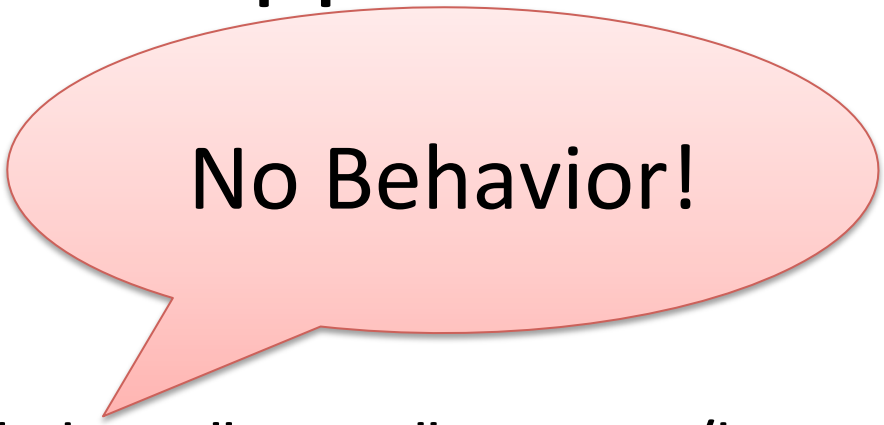


HTML File = App's View

```
...
<button id="resetButton" class="
<div id="simulationView">
...
<script src="js/matter.js"
<script src="js/index.js"></script>
...
```



HTML File = App's View




```
...
<button id="resetButton" class="green">Reset</button>
<div id="simulationView"></div>
...
<script src="js/matter.js"></script>
<script src="js/index.js"></script>
...
```

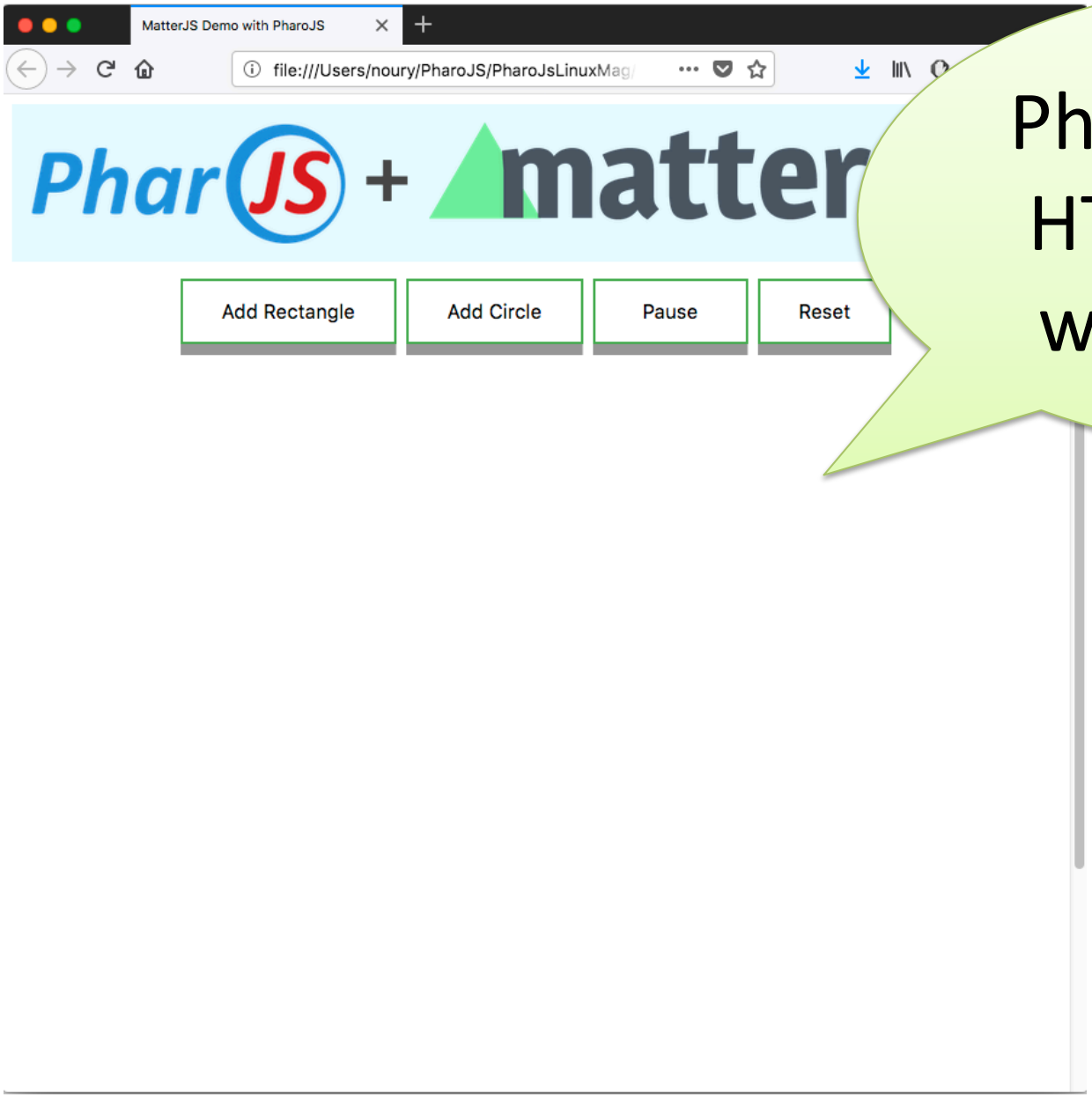
HTML File = App's View

No Behavior!

```
...
<button id="resetButton" class="green">Reset</button>
<div id="simulationView"></div>
...
<script src="js/matter.
<script src="js/ind
...
```

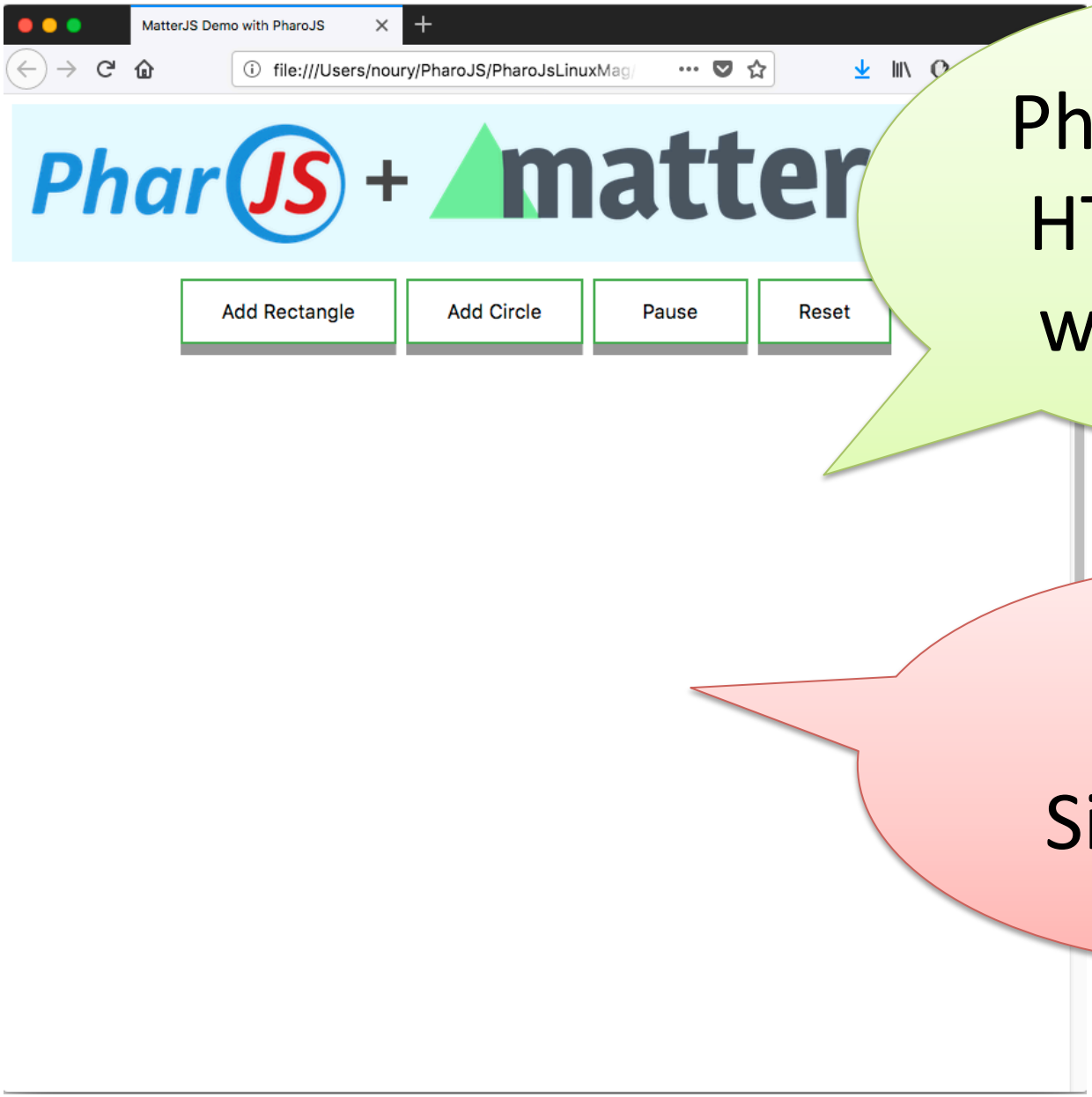
ID = to attach
Pharo code

Rerun Tests



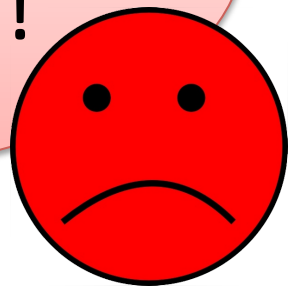
PharoJS Opens HTML file in a web browser

Rerun Tests

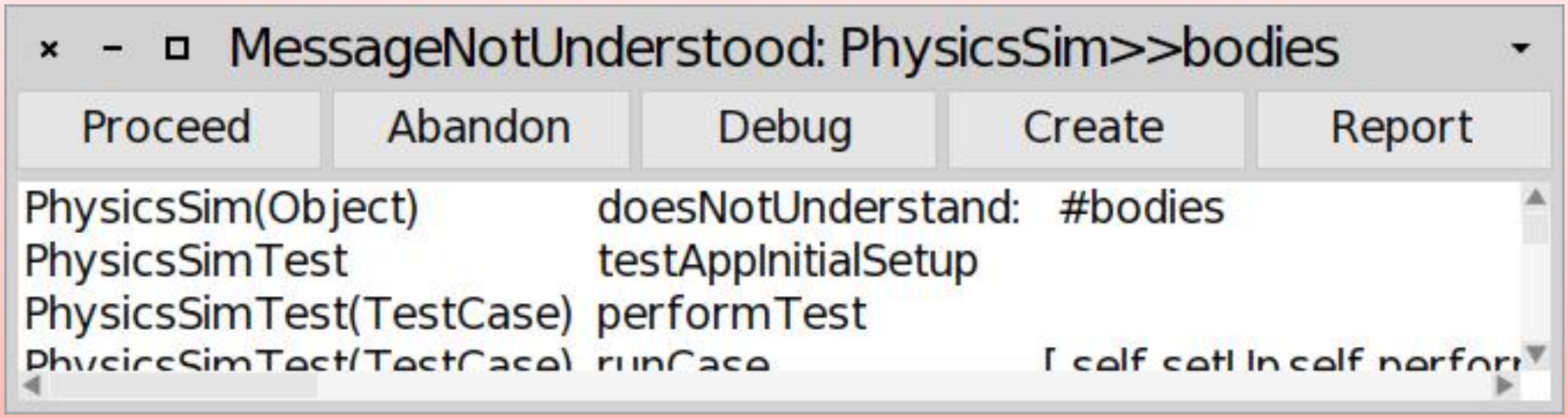


PharoJS Opens HTML file in a web browser

No Simulation!

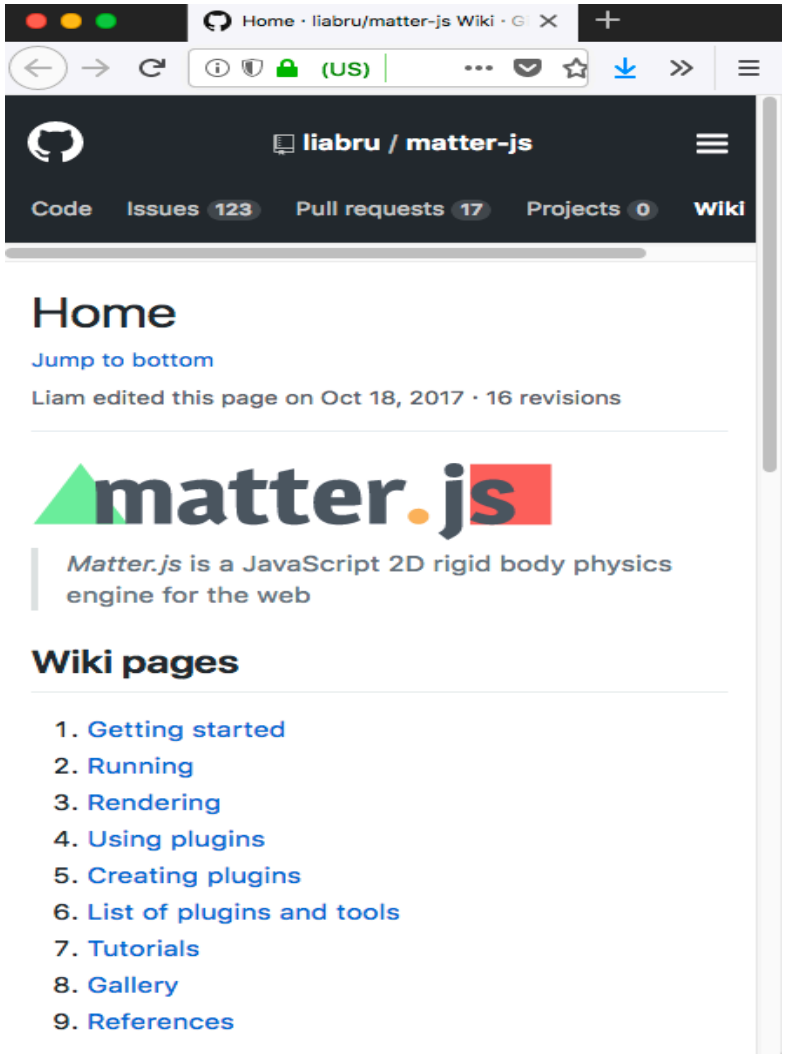


Rerun Tests


MessageNotUnderstood: PhysicsSim>>bodies
Proceed
Abandon
Debug
Create
Report
PhysicsSim(Object) doesNotUnderstand: #bodies
PhysicsSimTest testAppInitialSetup
PhysicsSimTest(TestCase) performTest

Fixing the App

JS Doc/Examples → Pharo Code



PhysicsSim>>initialize

```
super initialize.
self createAndStartEngine.
self setupAndStartRendering.
```

PhysicsSim>>bodies

```
^ engine world bodies
```

Fixing the App

PhysicsSim>>matterJsRoot

^ window Matter

PhysicsS createAndStartEngine

| r

ine create.

ner create.

: runner with: engine

Proxy to
Javascript global

Fixing the App

PhysicsSim>>matterJsRoot

```
^ window Matter
```

PhysicsSim>>createAn

```
| runner |
engine := self matterJs
runner := self matterJsRoot Runner create.
self matterJsRoot Runner run: runner with: engine
```

PharoJS generated accessor to JS object property

Fixing the App

```
PhysicsSim>>matte
    ^ window Mat

PhysicsSim>>createAn
    | runner |
    engine := self matterJsRoot Engine create.
    runner := self matterJsRoot Runner create.
    self matterJsRoot Runner run: runner with: engine
```

Calling JS method
create()

Fixing the App

```
PhysicsSim>>matterJsRoot
    ^ window Matter

Phy
    | r
    engine
    runner := self matterJsRo nner create.
    self matterJsRoot Runner run: runner with: engine
```

Calling JS method
run(*runner, 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:
        {#
        #
        #
                                                    bject.
self matterJsRoot Render run: render
```

HTML

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

Fixing it

Create a JS Object from a Pharo Dictionary

```
PhysicsSim>>se
| render simulatio
simulationView := self elem                    #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

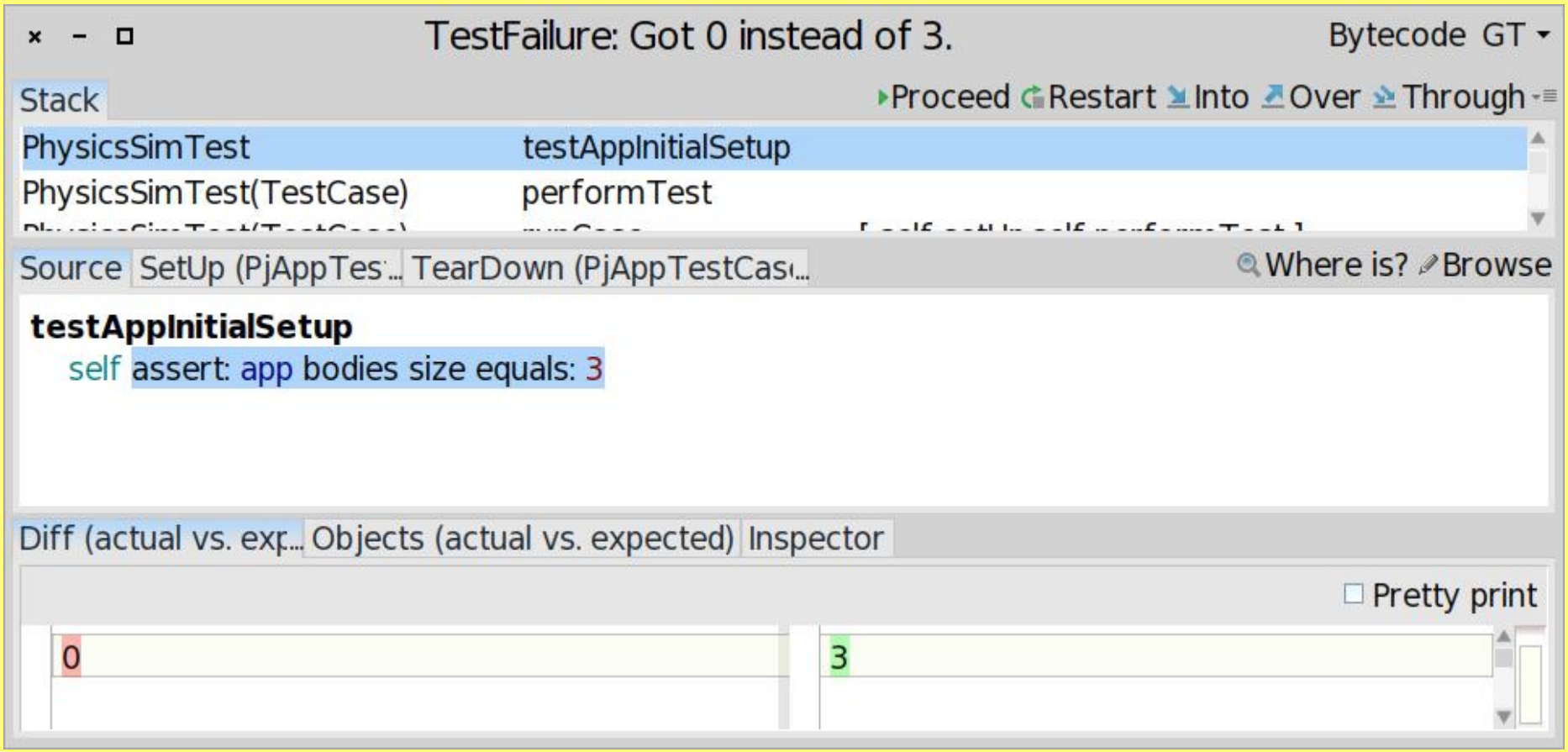
```
{element : simulationView.
engine :engine.
options : {
    width : 800.
    height : 600.
    wireframes : false.
    background : 'transparent'}
}
```

Pharo Code

Pharo equivalent

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

Rerun Tests


TestFailure: Got 0 instead of 3.
Bytecode GT
Stack
Proceed
Restart
Into
Over
Through
PhysicsSimTest
testAppInitialSetup
PhysicsSimTest(TestCase)
performTest
Source
SetUp (PjAppTes...
TearDown (PjAppTestCas...
Where is?
Browse
testAppInitialSetup
self assert: app bodies size equals: 3
Diff (actual vs. exp...
Objects (actual vs. expected)
Inspector
Pretty print
0
3

Fixing the App

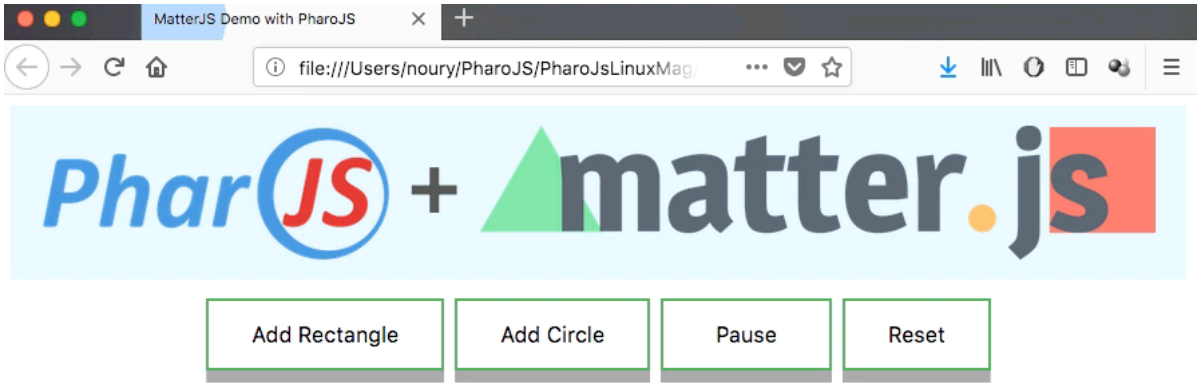
PhysicsSim>>initialize

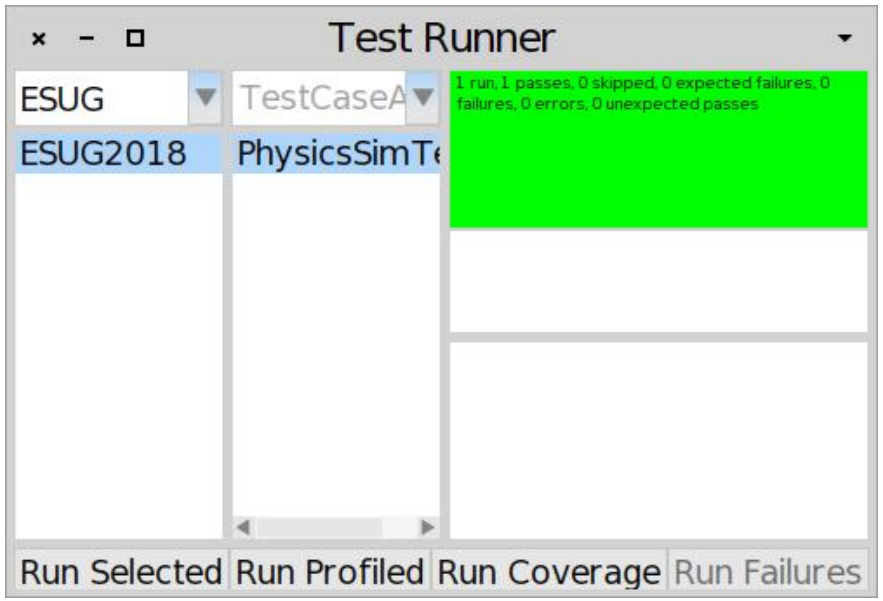
```
super initialize.
self createAndStartEngine.
self setupAndStartRendering.
self addInitialBodies
```

Creating MatterJS Bodies

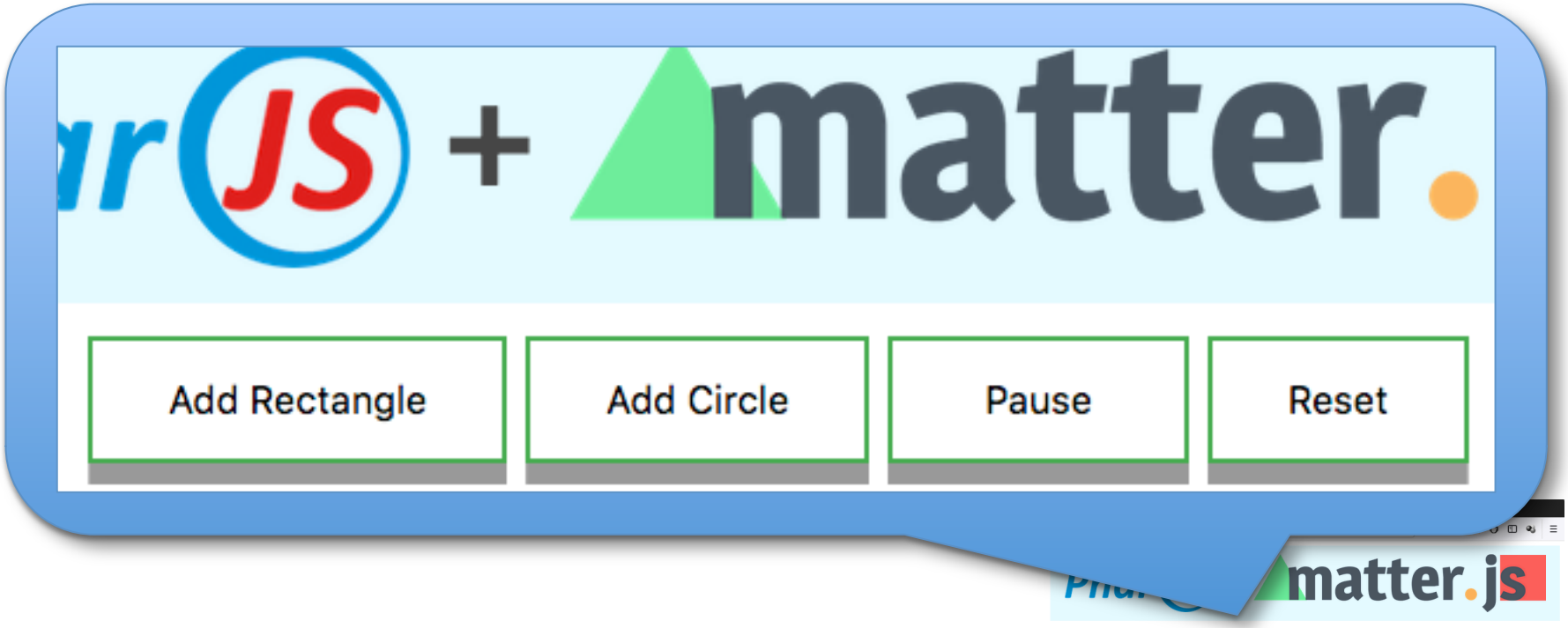
JS Doc/Examples	Pharo Code
Javascript code	**Pharo equivalent**
Bodies.rectangle(400, 610, 810, 60, { isStatic: true });	self matterJsRoot Bodies rectangle: 400 y: 610 width: 810 height: 60 options: { #isStatic -> true} **asJsObject**.

1st Test Passes!

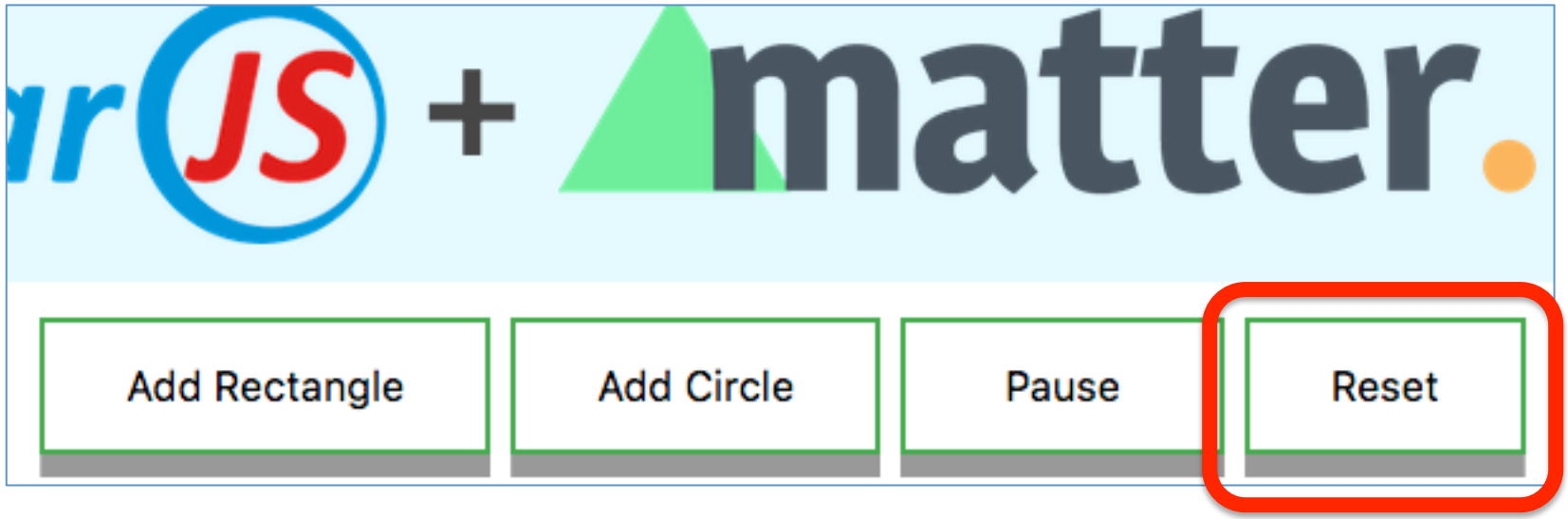




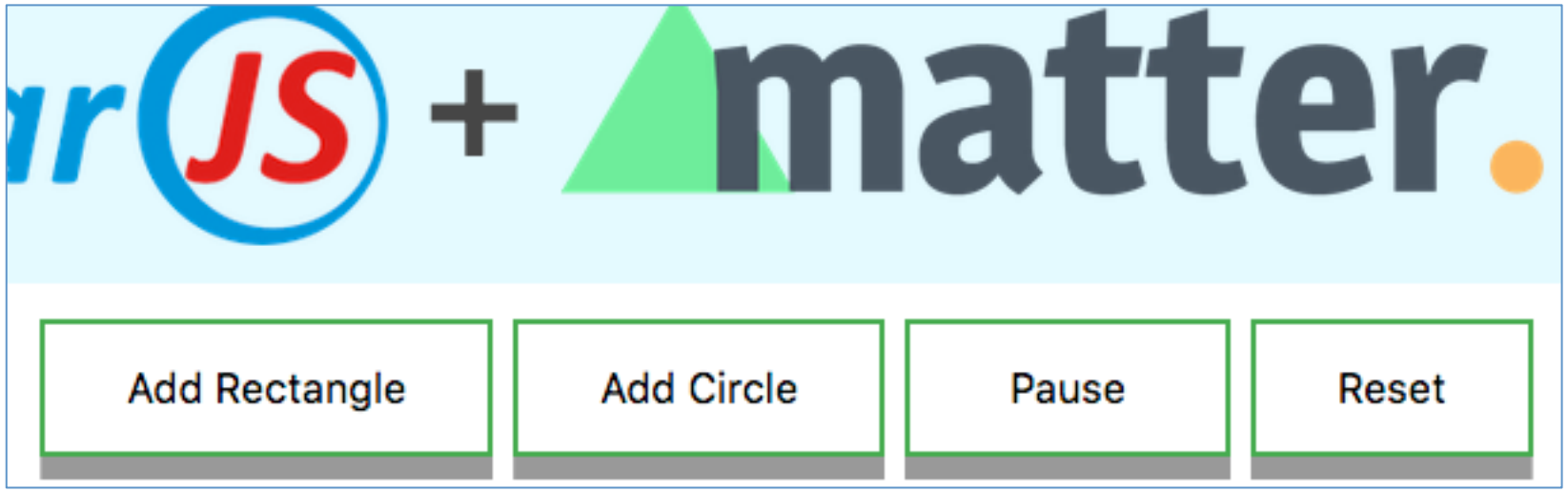
More features = Button Handling



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

<u>PhysicsSimTest>>testResetDeleteMobileBodiesOnly</u>

self clickElementById: #resetButton.

self assert: **ap** ... equals: **1**.

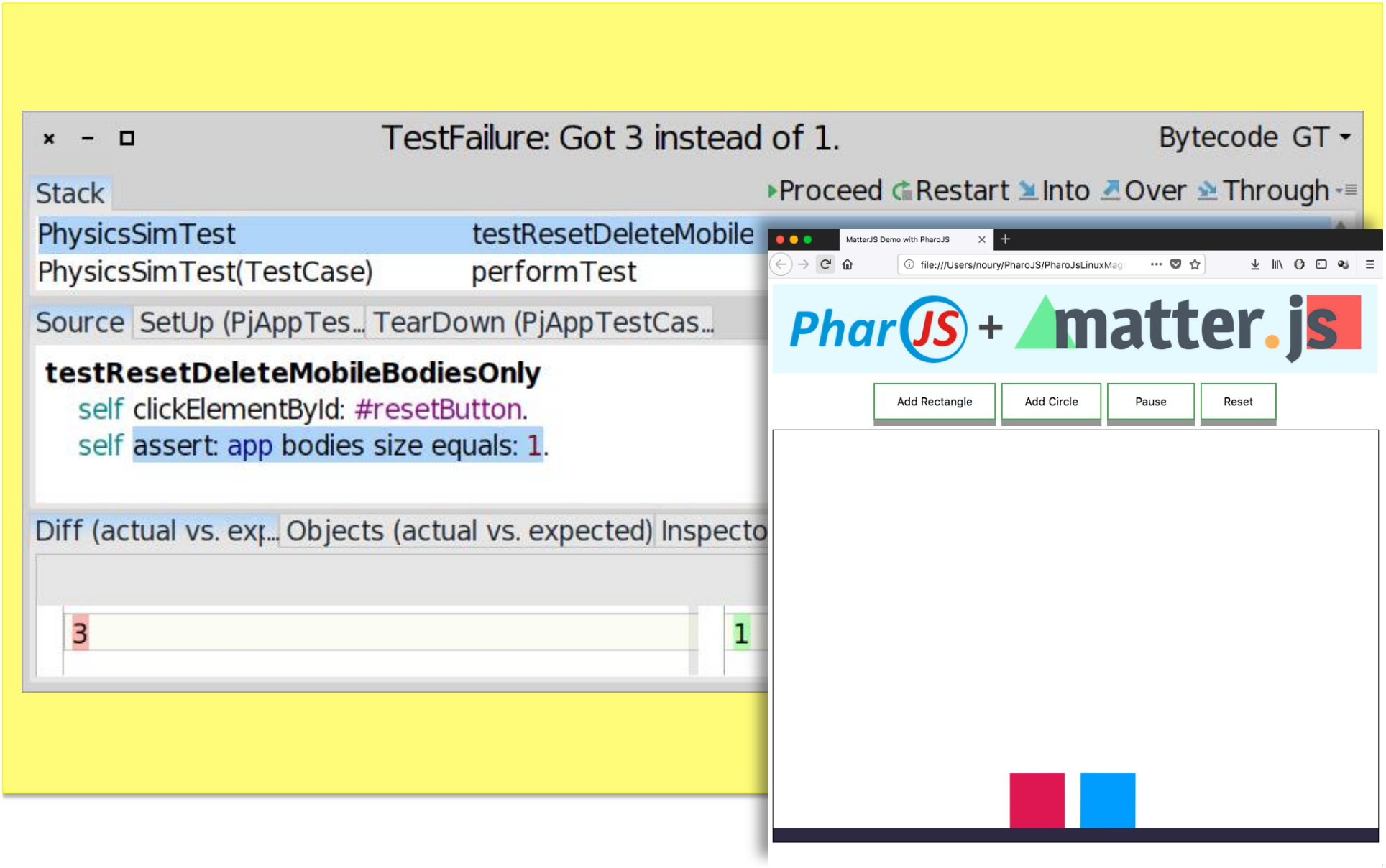
…

<button ... ton>

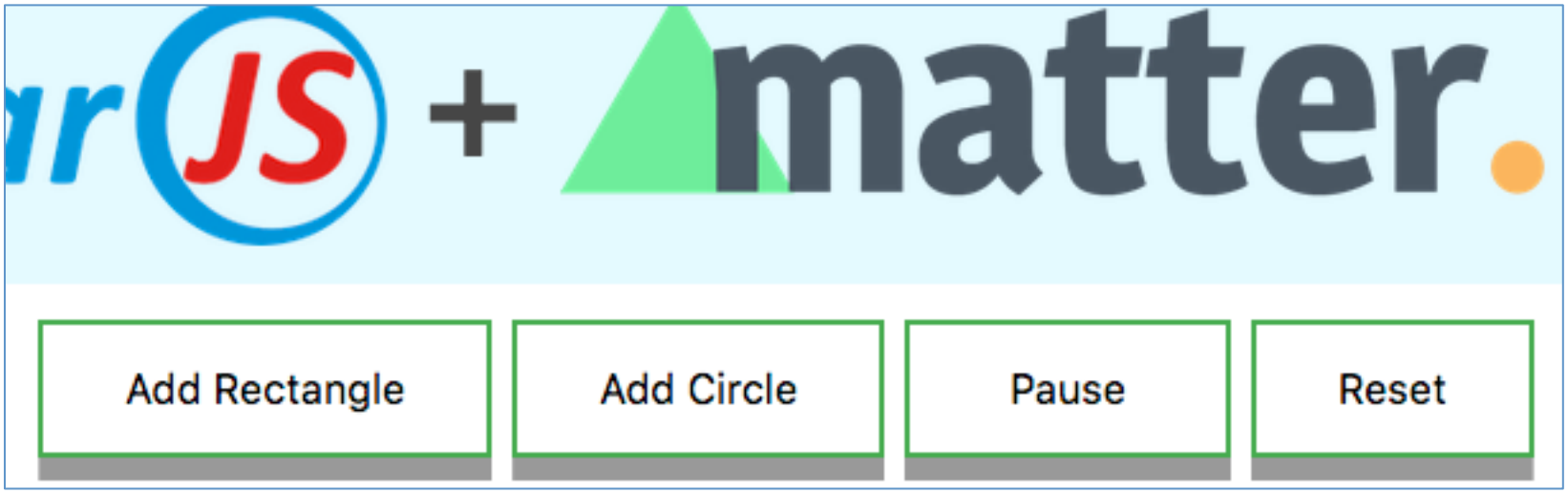
…

Pharo triggers event on DOM object!

Test Fails


TestFailure: Got 3 instead of 1.
Bytecode GT
Stack
Proceed
Restart
Into
Over
Through
PhysicsSimTest
testResetDeleteMobile
PhysicsSimTest(TestCase)
performTest
Source
SetUp (PjAppTes...
TearDown (PjAppTestCas...
testResetDeleteMobileBodiesOnly
self clickElementById: #resetButton.
self assert: app bodies size equals: 1.
Diff (actual vs. exp...
Objects (actual vs. expected)
Inspecto
3
1
MatterJS Demo with PharoJS
file:///Users/noury/PharoJS/PharoJsLinuxMag
PharJS + matter.js
Add Rectangle
Add Circle
Pause
Reset

Add Behavior to HTML Button



PhysicsSim>>initialize

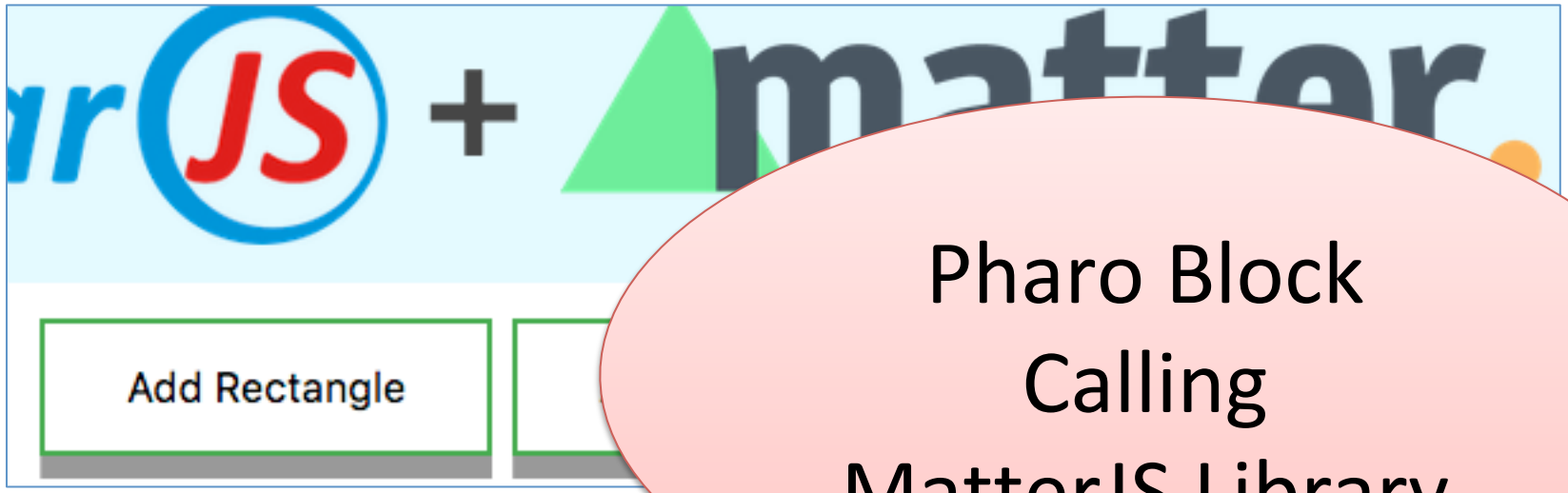
```
...
self onClick: #resetButton
    do: [ self matterJsRoot World
            clear: engine world
            keepStatic: true]
```

Add Behavior to HTML Button

PhysicsSim>>initialize

```
...
self onClick: #resetButton
    do: [ self matterJsRoot World
            clear: engine world
            keepStatic: true]
```

Add Behavior to HTML Button

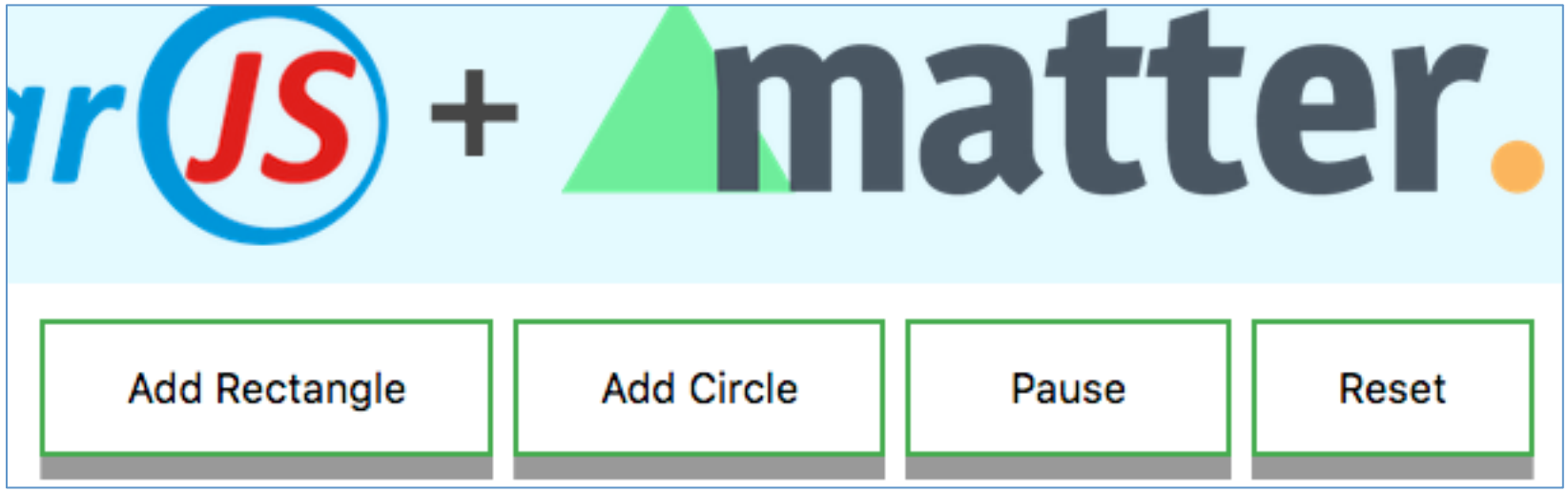


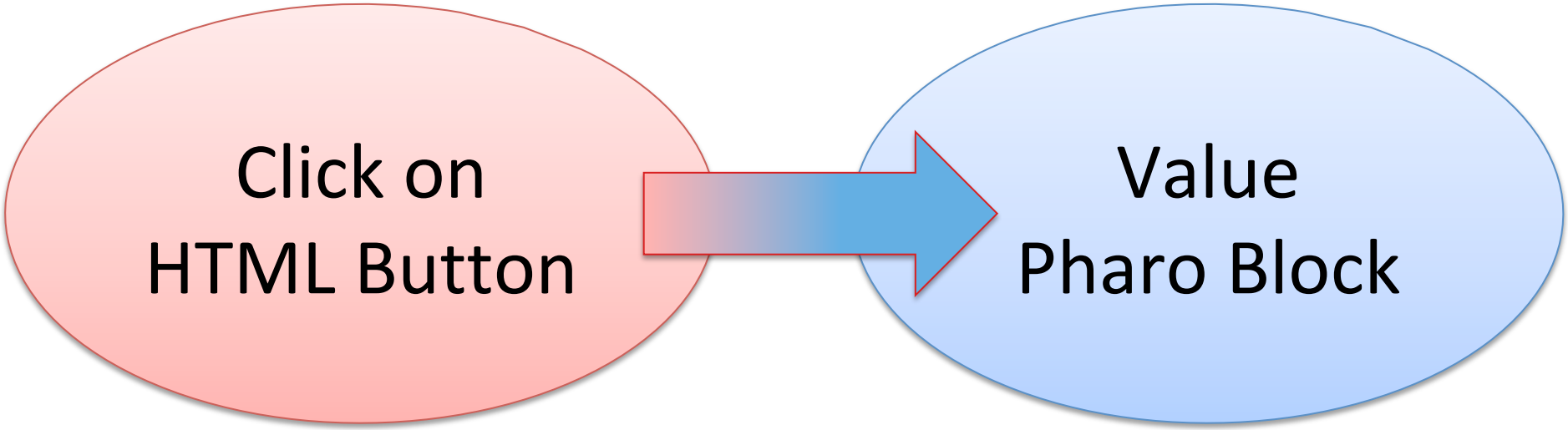
Pharo Block
Calling
MatterJS Library

PhysicsSim>>initialize

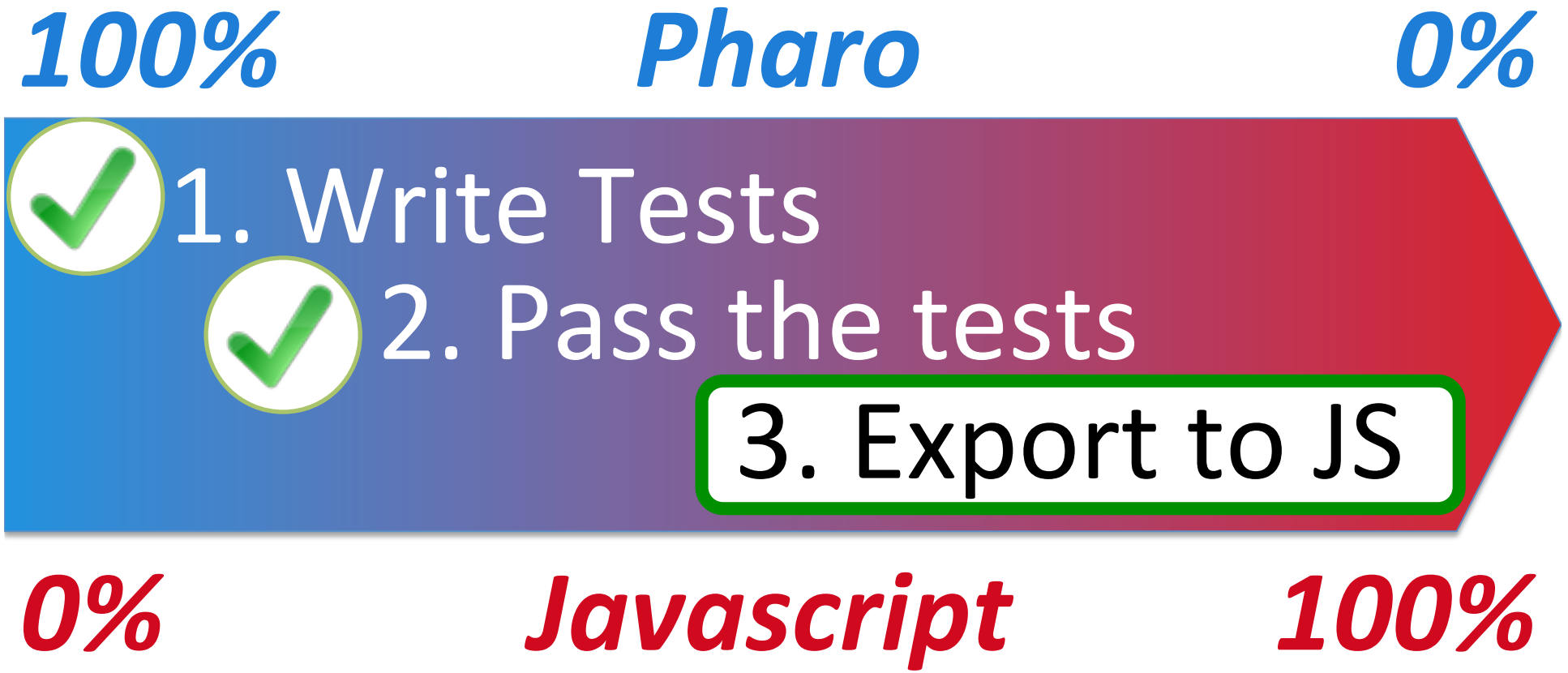
```
...
self onClick: #resetButton
    do: [ self matterJsRoot World
            clear: engine world
            keepStatic: true]
```

Add Behavior to HTML Button



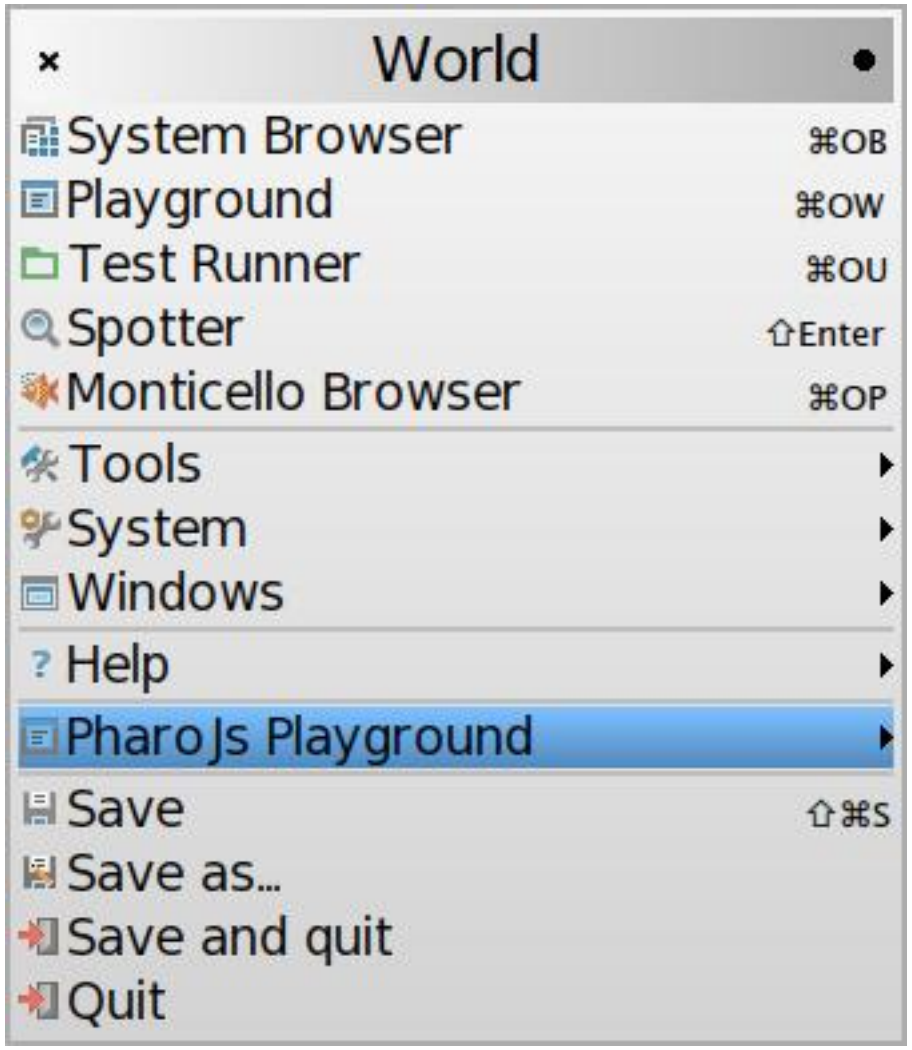


Lifecycle with PharJS



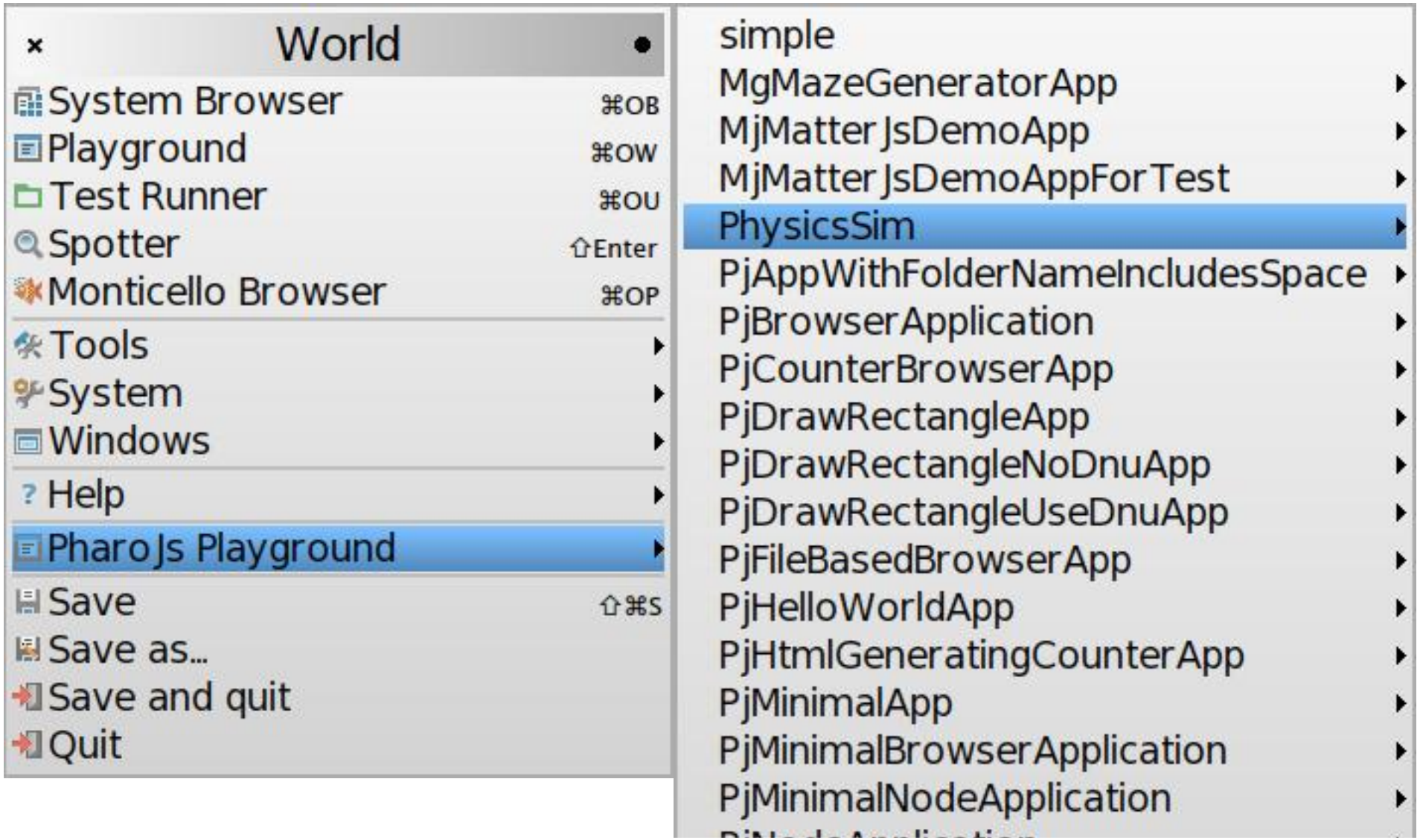
Export App for Production

100% Javascript



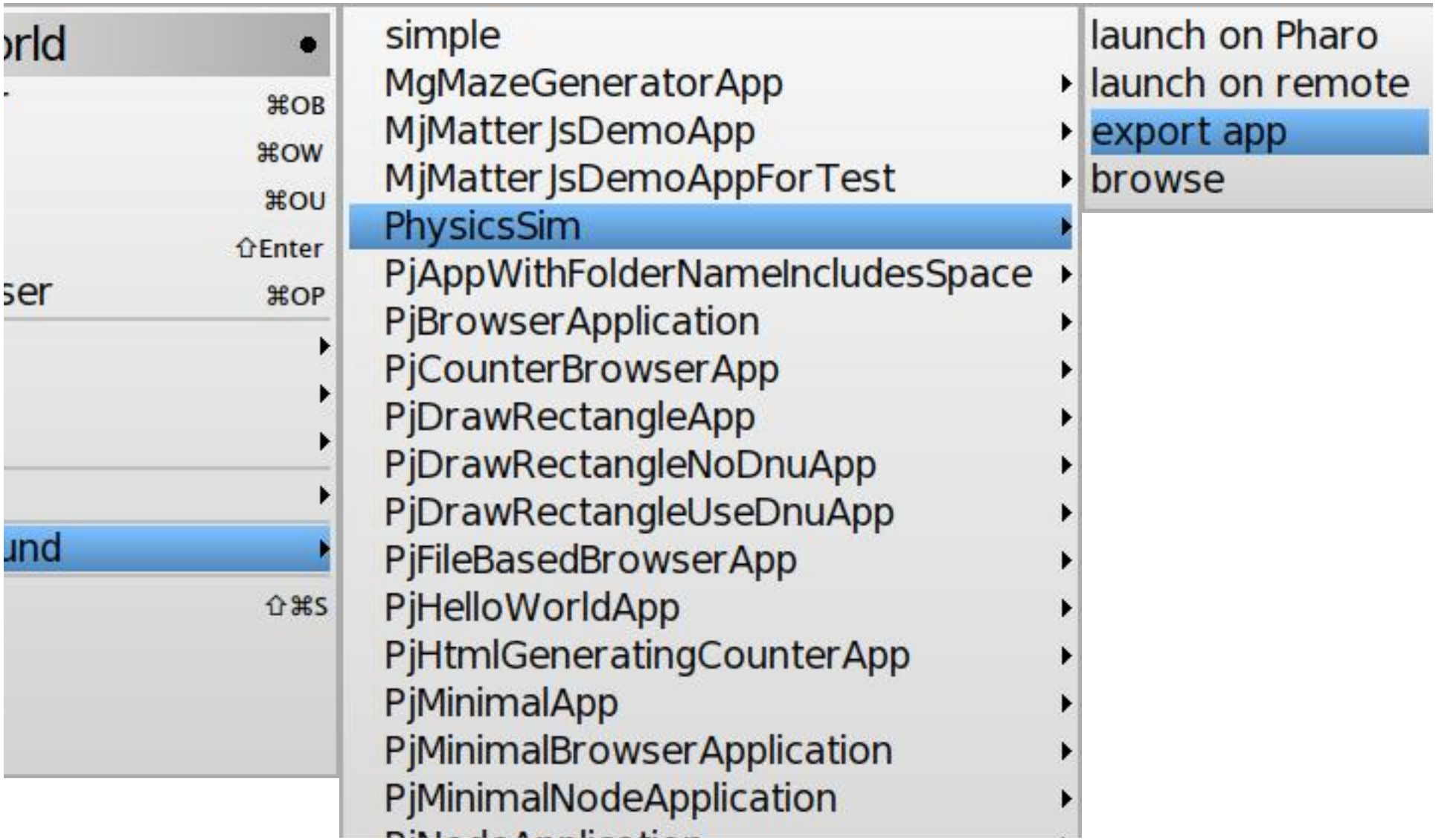
Export App for Production

100% Javascript

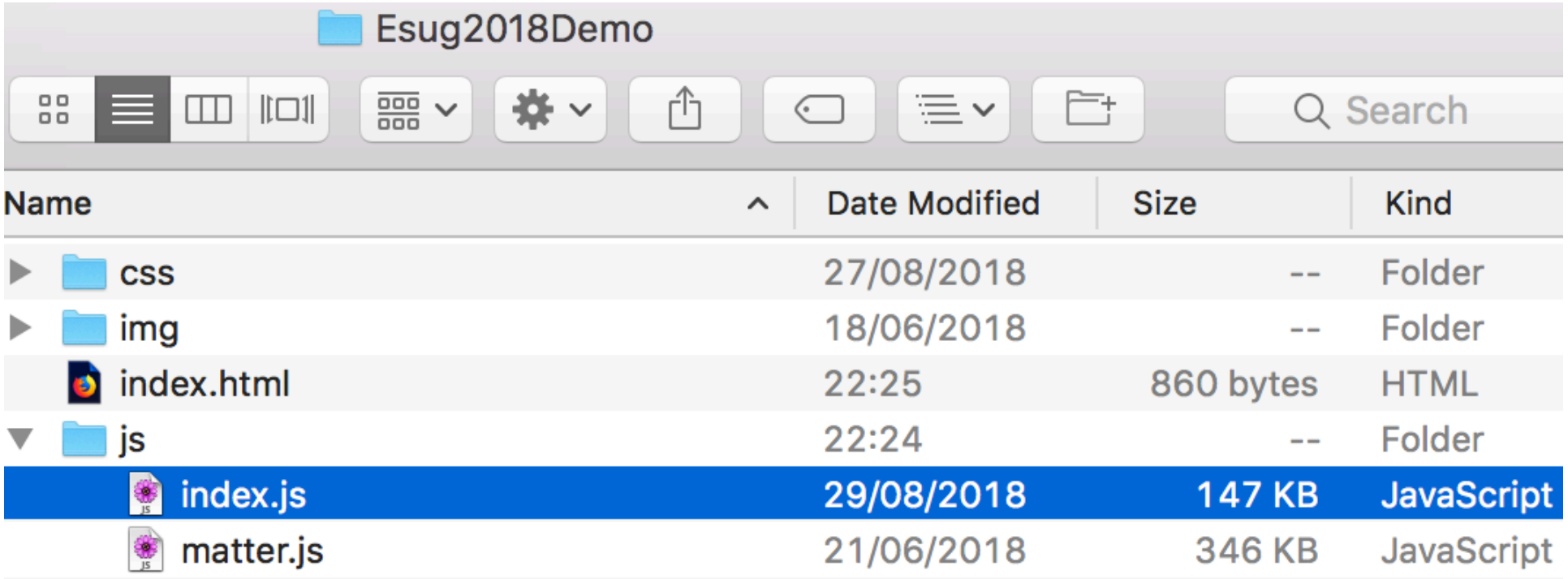


Export App for Production

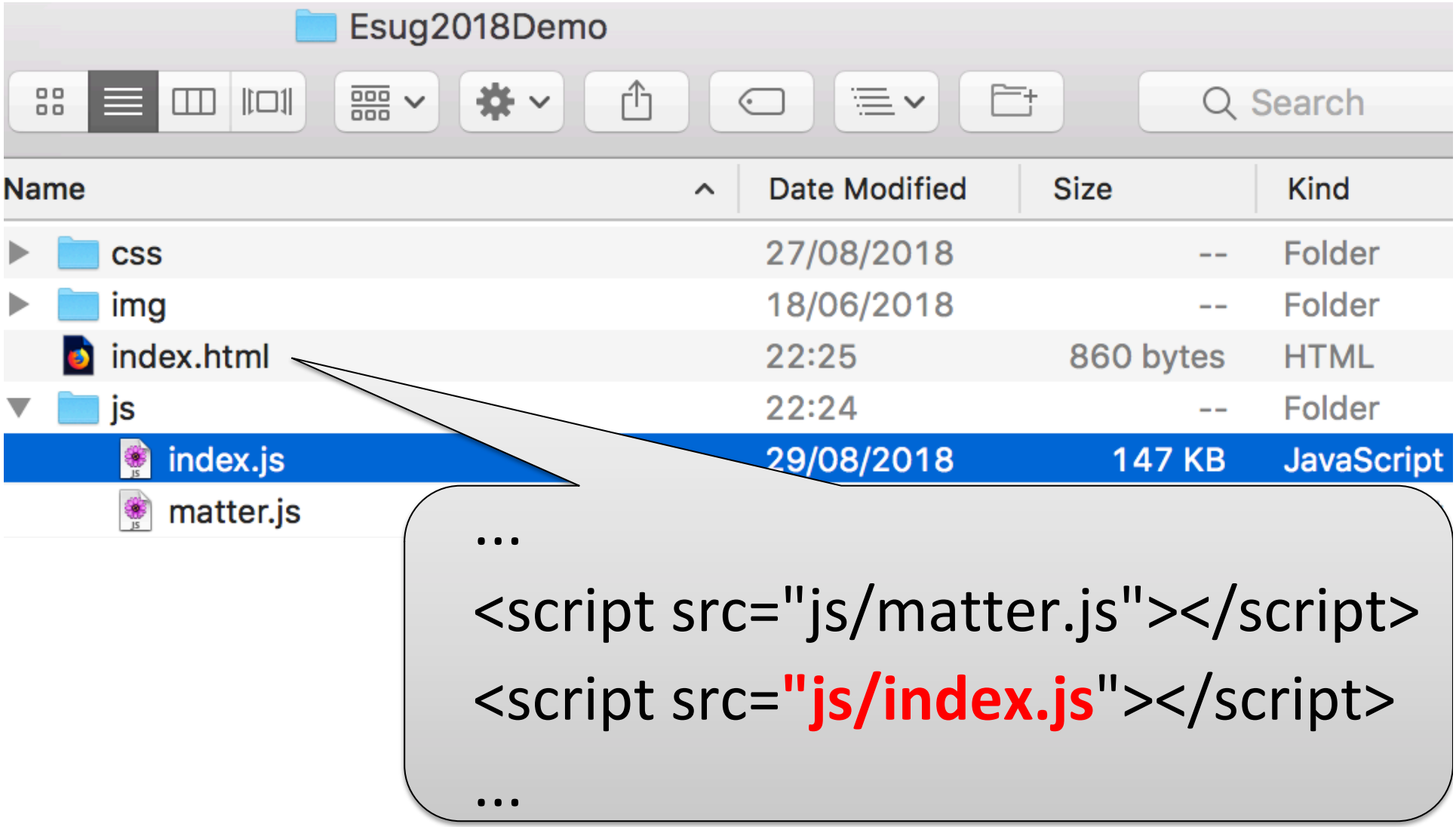
100% Javascript



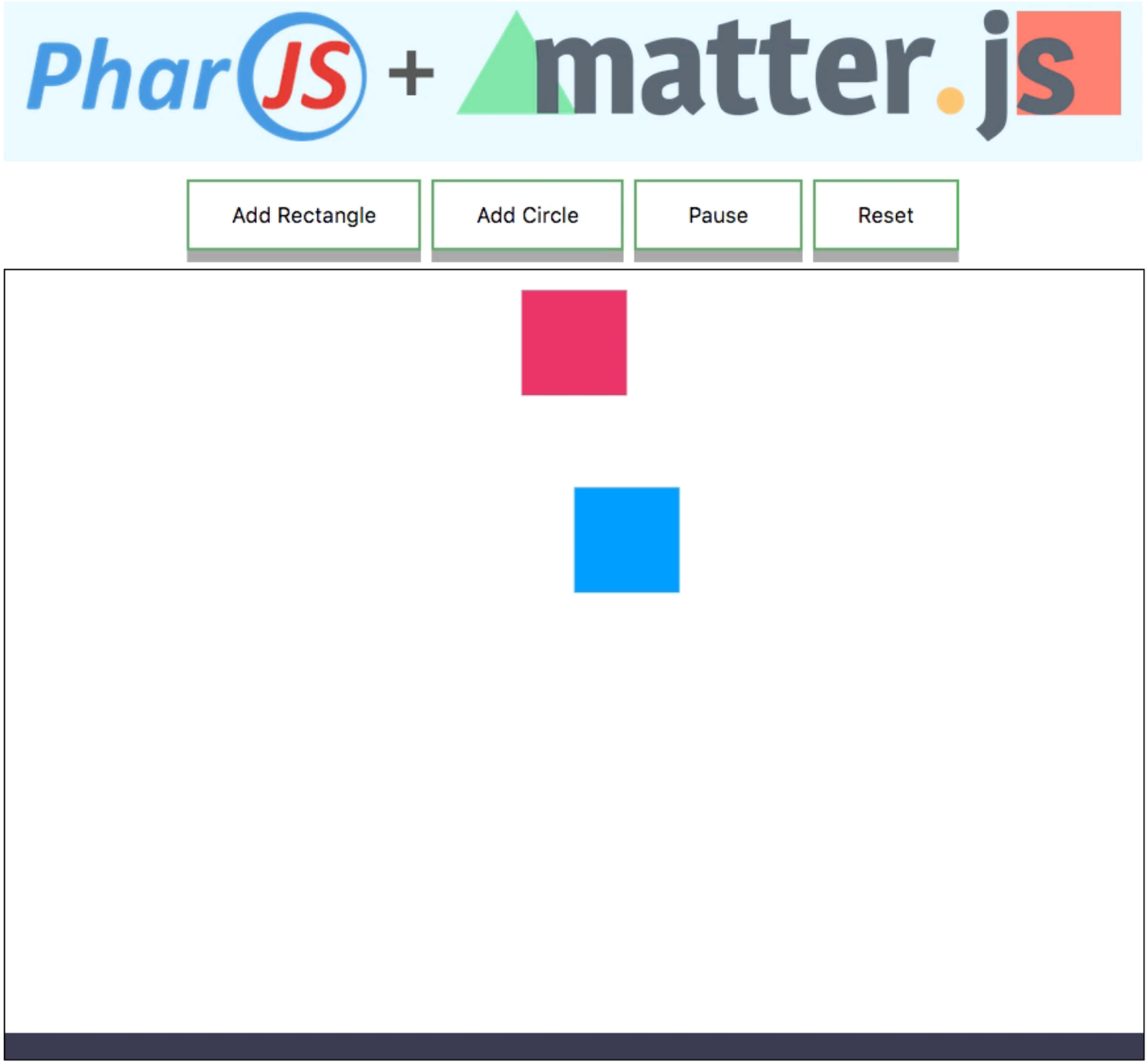
Exported JS file



Exported JS file



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



Beyond Research!

Open Source for the Real World

Pharo Tradition!

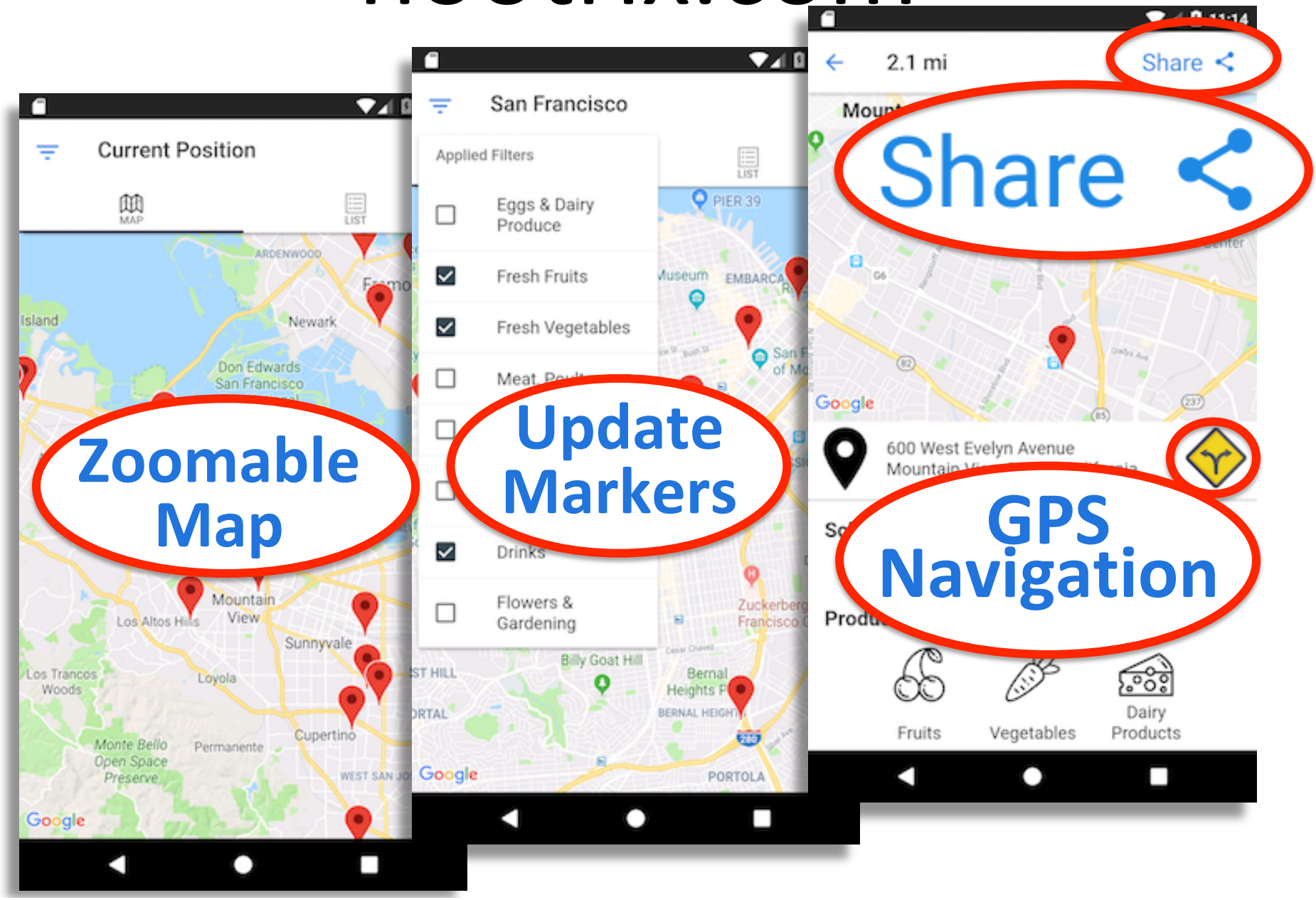
nootrix.com

iOS

ANDROID

powered by PharJS

nootrix.com


Current Position
MAP
LIST
Zoomable Map
San Francisco
Applied Filters
Eggs & Dairy Produce
Fresh Fruits
Fresh Vegetables
Drinks
Flowers & Gardening
Update Markers
2.1 mi
Share
Share
600 West Evelyn Avenue
GPS Navigation
Fruits
Vegetables
Dairy Products
Google

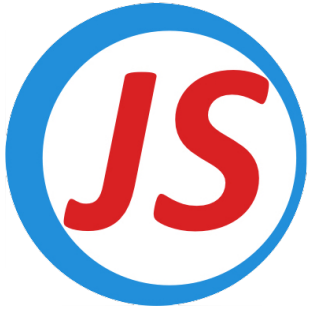
What Next?

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

This afternoon!

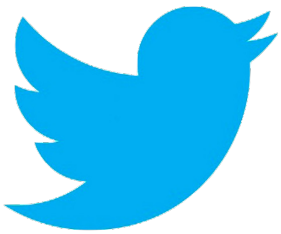
Learn more about PharoJS

- 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