Pharo 9

Meeting the XXI century

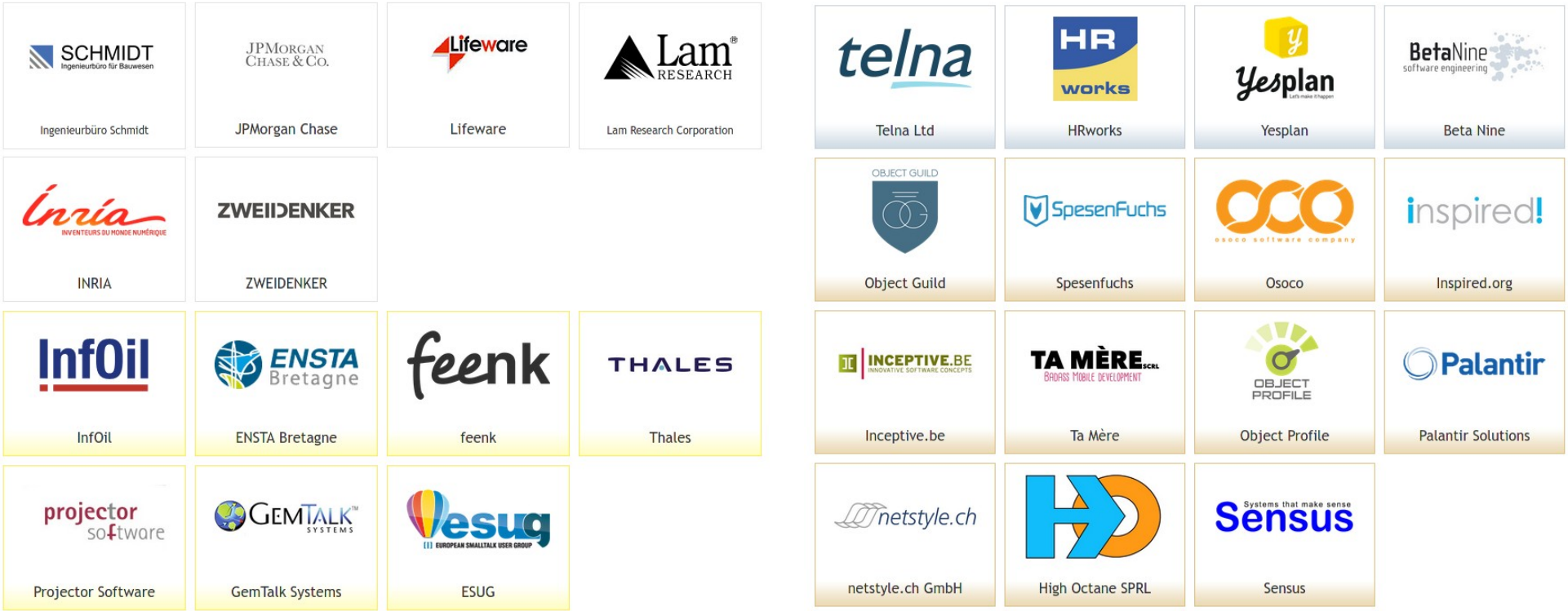
Esteban Lorenzano

- Pharo consortium engineer since 2018
- Pharo architect since 2012
- Owned a company to develop in Pharo back in 2008
- Java senior architect for 7 years (and 15 years overall java experience)
- Web, microprocessors, etc., etc., etc.
- JavaScript, C++, ObjC, C#, Delphi, ASM and lots of languages that no longer exist or have been long-time forgotten
- 28 years (!) programming experience

Pharo

Pharo's goal is to deliver a clean, innovative, free open-source Smalltalk-inspired environment. By providing a stable and small core system, excellent dev tools, and maintained releases, Pharo is an attractive platform to build and deploy mission critical Smalltalk applications.

Pharo Consortium



Industrial Members

Pharo Consortium

Academic Members

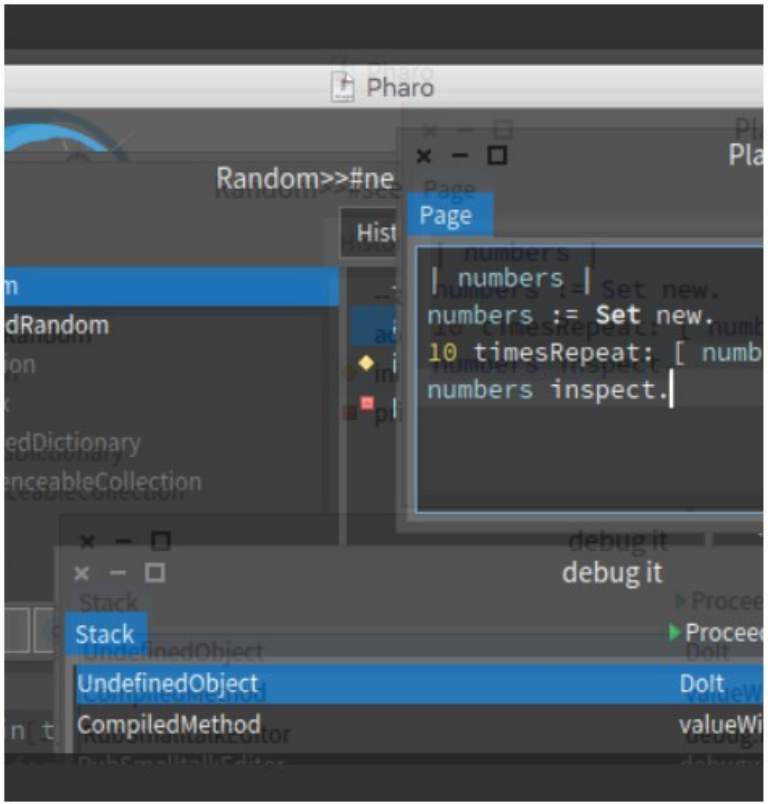
Pharo Association

Welcome to the Pharo Association

Welcome to the Pharo user association. Our goal is to help individuals support the promotion and development of Pharo. Indeed there is a Pharo consortium for legal entities such as companies, universities, associations and the Pharo association. These are two different things: the association is for individuals and the consortium for legal entities. The association is a french *Loi de 1901* association.

The driving idea behind our association is that you as a single individual can have a definite impact as part of our group. To support Pharo, just join us and contribute to shape the future.

The goal of Pharo is to deliver a clean, pure, reflective object-oriented language inspired by Smalltalk. Our second objective is to create an ecosystem composed of universities, research institutes and teams as well as companies.



Pharo 9 « simplified » roadmap

- New Tools: Playground, Inspector, Debugger, Spotter
- Spec2 (Morphic Back-end, Gtk Back-end)
- Enhance language: Full block closures, Fluid class builder, Microdown. H-completion, …
- Support for large images
- VM
  - Headless
  - (T)FFI
  - ARMv8

Pharo 9 : Some numbers

- 1174+ issues, 1925+ PRs closed since Pharo 8
- Not including all the work at :
  - pharo-project/opensmalltalk-vm
  - pharo-spec/Spec
  - pharo-spec/NewTools
  - … and many others
- 242 forks in GitHub

Pharo 9 : Some numbers

- Do not underestimate the value of education !
  - MOOC : 10000+ registrations
  - Lots of free Books & new Booklets
- A growing and welcoming community :
  - Discord : 2762 members, 382 online at the moment of taking this snapshot.
  - Discord : newbie, help, MOOC channels
  - Discord : language channels : es, fr, kmer, ua, zh
  - Lists pharo-users/pharo-dev (~600 users each)

Now, let’s see Pharo 9 in action :)

New Tools

- Pharo 9 goal : replace tools made with Glamour.
  - Debugger
  - Inspector
  - Playground
  - Spotter
- Allow the next step : switching back-end.
- We do not want just to replace but to improve.

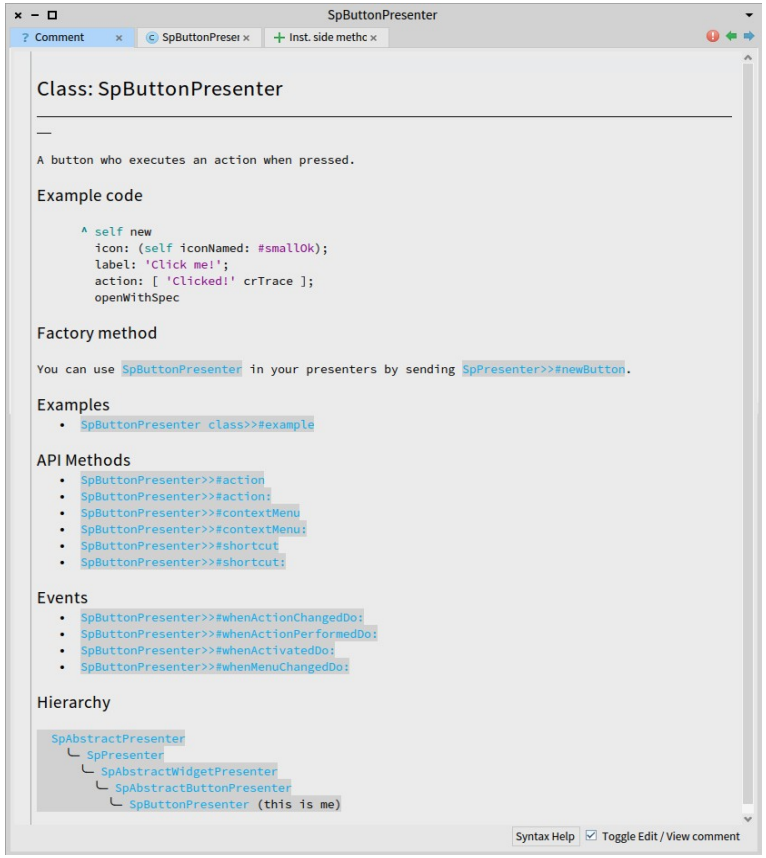
Spec 2

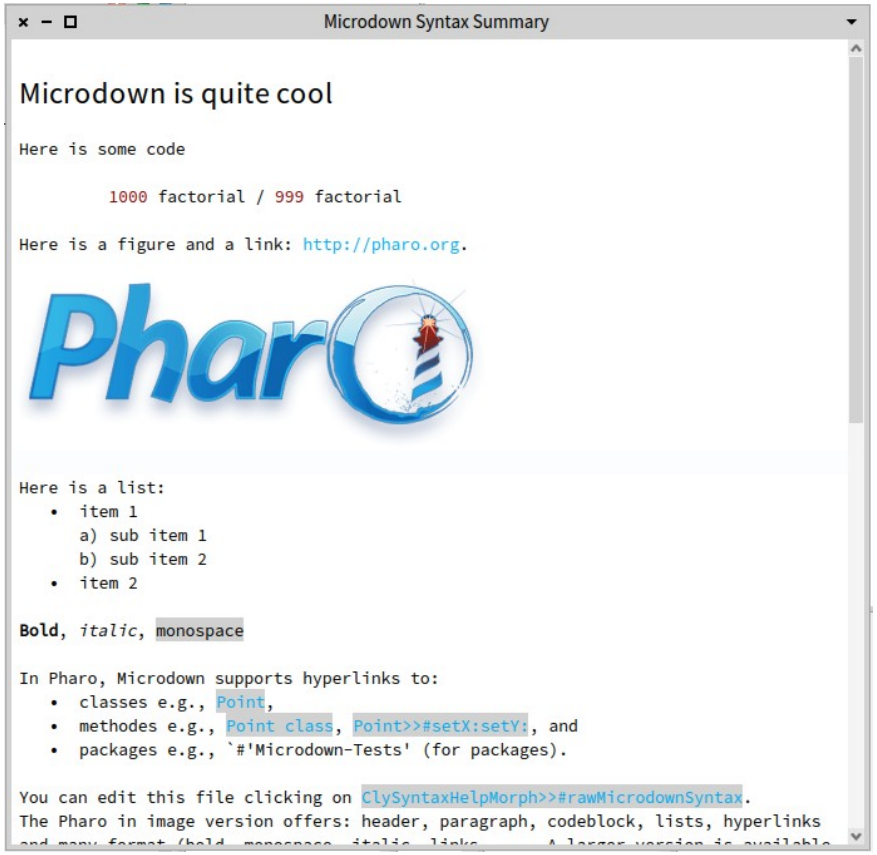
- We need an application building framework suitable for doing real desktop applications.
- We need also to be able to write the Pharo tools.
- We want to be able to switch back-ends
  - We do not want to rewrite all tools again if we decide to switch back-ends again in the future.
    - If technology gets old.
    - To add native back-ends.
  - Deprecate Morphic !

Fluid Class Builder

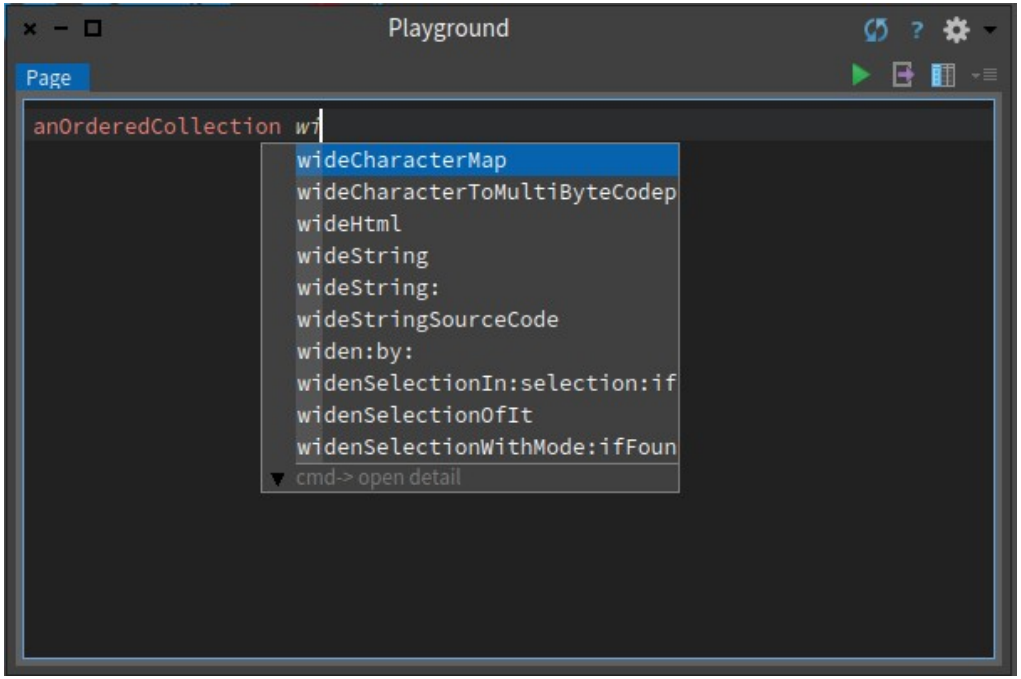
```
SpAbstractWidgetPresenter << #SpAbstractButtonPresenter
    slots: {
            #icon => SpObservableSlot .
            #label => SpObservableSlot };
    tag: 'Widgets';
    package: 'Spec2-Core'
```

Microdown

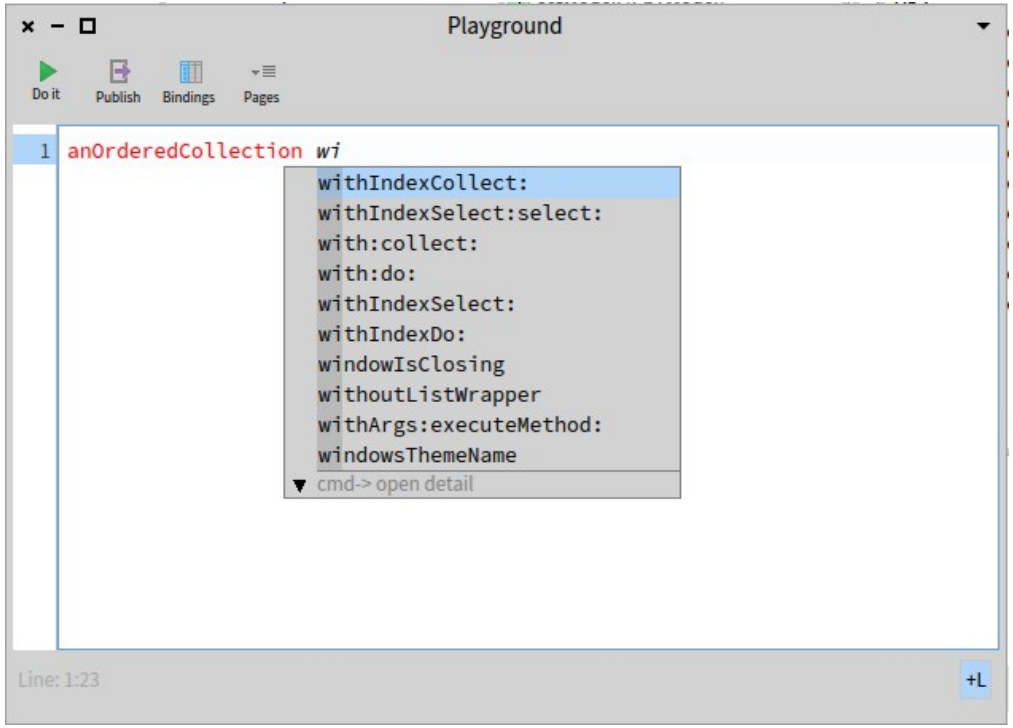

SpButtonPresenter
Comment
SpButtonPreser
Inst. side methc
Class: SpButtonPresenter
A button who executes an action when pressed.
Example code
^ self new
icon: (self iconNamed: #smallOk);
label: 'Click me!';
action: ['Clicked!' crTrace];
openWithSpec
Factory method
You can use SpButtonPresenter in your presenters by sending SpPresenter>>#newButton.
Examples
SpButtonPresenter class>>#example
API Methods
SpButtonPresenter>>#action
SpButtonPresenter>>#action:
SpButtonPresenter>>#contextMenu
SpButtonPresenter>>#contextMenu:
SpButtonPresenter>>#shortcut
SpButtonPresenter>>#shortcut:
Events
SpButtonPresenter>>#whenActionChangedDo:
SpButtonPresenter>>#whenActionPerformedDo:
SpButtonPresenter>>#whenActivatedDo:
SpButtonPresenter>>#whenMenuChangedDo:
Hierarchy
SpAbstractPresenter
SpPresenter
SpAbstractWidgetPresenter
SpAbstractButtonPresenter
SpButtonPresenter (this is me)
Syntax Help
Toggle Edit / View comment


Microdown Syntax Summary
Microdown is quite cool
Here is some code
1000 factorial / 999 factorial
Here is a figure and a link: http://pharo.org.
Here is a list:
item 1
a) sub item 1
b) sub item 2
item 2
Bold, italic, monospace
In Pharo, Microdown supports hyperlinks to:
classes e.g., Point,
methodes e.g., Point class, Point>>#setX:setY:, and
packages e.g., `#'Microdown-Tests' (for packages).
You can edit this file clicking on ClySyntaxHelpMorph>>#rawMicrodownSyntax.
The Pharo in image version offers: header, paragraph, codeblock, lists, hyperlinks

Heuristic Completion



Pharo 8



Pharo 9

Parser improvements

```
example

    | presenter |
    ))
    ((
    (presenter := SpPresenter new)
        layout: (SpBoxLayout newVertical
            add: (presenter newButtonBar
                add: presenter newButton;
                )).
                ^ true.
                ^ true
```

Pharo 8

```
example

    | presenter |
    ))
    ((
    (presenter := SpPresenter new)
        layout: (SpBoxLayout newVertical
            add: (presenter newButtonBar
                add: presenter newButton;
                )).
                ^ true.
                ^ true
```

Pharo 9

Support for large images

We developed an « all in image » solution, and we have validated our approach building a large image generator.

With it we test our infrastructure on generated images with around ~1.500.000 methods and ~95.000 classes, including long methods with big selectors and literals.

The indexing system presents a memory footprint of around 20%, but it allows us to search the image without pauses. Improving the whole experience.

Being used by a company with a HUGE code base

(T)FFI

- Using **libffi** as backend (working on ARMv8 and M1)
- Different strategies for different needs
  - Same-Thread strategy (same performance as before)
  - Worker strategy
  - Main thread strategy

Virtual Machine

- We took control because :
  - We cannot depend on just one single person.
  - We have plans that were being holded back.
  - We have different goals.

Virtual Machine – Actions

- 3000+ unit tests added
  - Regression detection
- Ephemerons working for real
- ARMv8
  - Windows
  - Raspbian
  - M1
- Knowledge gained by the Pharo community

Headless (idle) VM

- It is really headless (no hidden window as before)
  - Allows embedding
  - Still some minimal changes needed in image (likely for Pharo 10)
- It sleeps when no events are being processed
  - Suitable for embed into another applications (but I/O still needs to be tweaked)
  - Suitable for server-side.
  - Suitable for not-killing your laptop battery :)

VM - Others

- Primitive speed improprement in Windows.
- Repeatable building process.
- AIO improvements (using modern function implementations).
- Unification of Socket implementation.
- Cleanups, updates.
- Documentation

Open Building Service

	Arch	Debian_10	Debian_9.0	Debian_Testing	Fedora_31	Fedora_32	Fedora_33	Raspbian_10		Raspbian_9.0	
	x86_64	x86_64	x86_64	x86_64	x86_64	x86_64	x86_64	aarch64	x86_64	aarch64	x86_64
libffi7		succeeded	succeeded		succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded
libgit2-1		succeeded		failed							
pharo9	failed	succeeded	failed	failed	failed	failed	failed	succeeded	succeeded	failed	failed
pharo9-ui	succeeded	succeeded	succeeded	failed	succeeded	succeeded	succeeded		succeeded		succeeded

	Raspbian_9.0		openSUSE_Leap_15.1	openSUSE_Leap_15.2	openSUSE_Tumbleweed	xUbuntu_18.04	xUbuntu_19.04	xUbuntu_20.04	
	arch64	x86_64	x86_64	x86_64	x86_64	x86_64	x86_64	aarch64	x86_64
libffi7	:ceeded	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded	succeeded
libgit2-1			succeeded	succeeded		succeeded	succeeded		succeeded
pharo9	failed	failed	failed	failed	failed	failed	succeeded	succeeded	succeeded
pharo9-ui		succeeded	succeeded	succeeded	succeeded	succeeded	succeeded		succeeded

Pharo 9 is a solid step to the future !