Pharo 11:
A stabilization release

http://sducasseatwork@mailo.com
http://www.pharo.org

LABWARE
Results Count

esug
EUROPEAN SMALLTALK USER GROUP

Inria

Lam®
RESEARCH

SCHMIDT
Ingenieurbüro für Bauwesen

Lifeware

Yesplan
Let's make it happen

OBJECT GUILD

THALES
Building a future we can all trust

ZWEIDENKER

telna

GemTalk™
SYSTEMS

projector
software

inspired!

netstyle.ch

InfOil

FINWORKS
INFORMATION TECHNOLOGY SOLUTIONS

BetaNine
software engineering

TA MÈRE SCRL
BADASS MOBILE DEVELOPMENT

Systems that make sense
Sensus

feenk

cirad

MINES
Douai
IA
Informatique
et Automatique

FAST
Fundación Argentina de Smalltalk

u b
UNIVERSITÄT
BERN

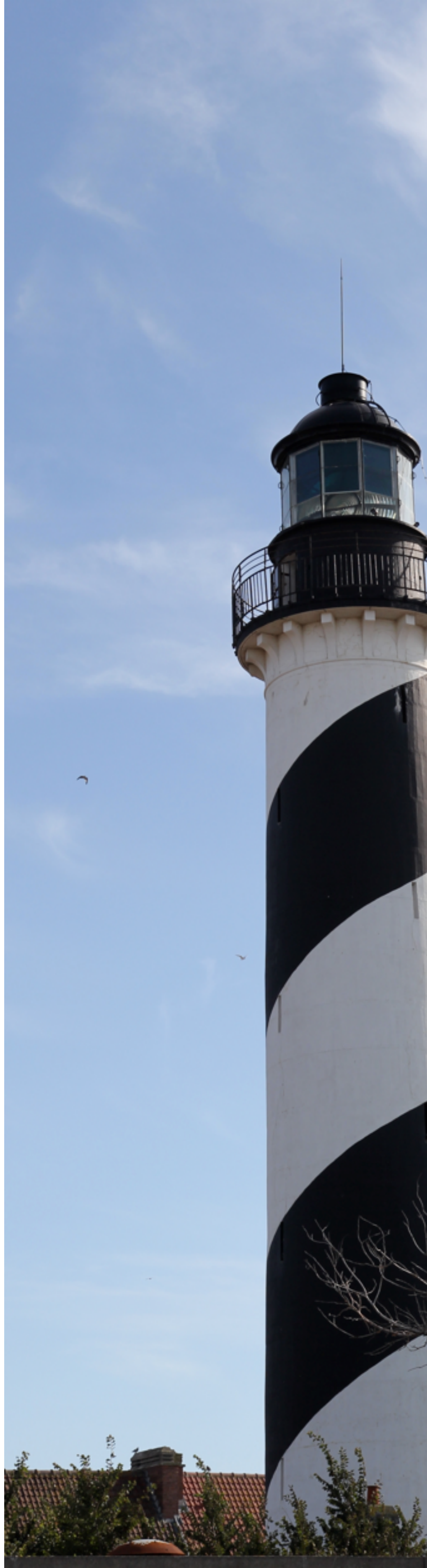
Toronto
Metropolitan
University

CZECH TECHNICAL UNIVERSITY IN PRAGUE

SRA
GROUP

project
uqbar

Evref

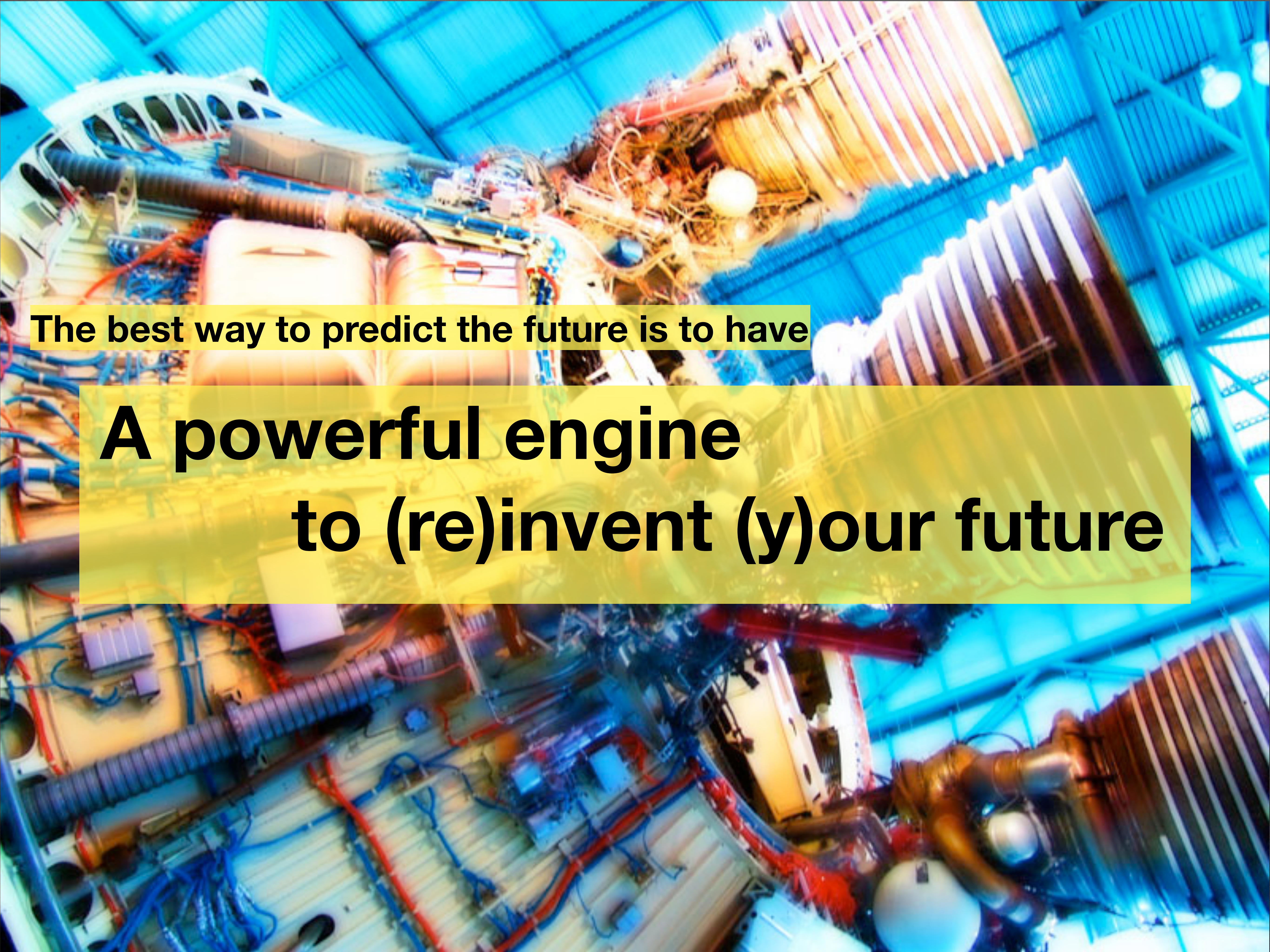


Talk Outline

- Pharo goals
- Pharo 11 and remarkable achievements :)
- Pharo 12 Preview

Pharo’s goals did not change

…



The best way to predict the future is to have

A powerful engine
to (re)invent (y)our future

Build/support/sustain

An ecosystem where innovation/business bloom

… that **you** can make money with Pharo.

… to be able to invent solutions to existing problems.

… a powerful innovative dynamic language where we can build (y)our future.

Super powerful live programming language and tools

A(n eco-)system
that can **evolve**

Each time we change something we think in terms of impact and support

trentosur
Soluciones móviles para retail y trade marketing
Nos enfocamos en lo que importa del negocio
PharoCloud
Pharo platform as a Service: put your Smalltalk web-application online at Pharocloud in just 3 clicks
Try it for FREE
Watch how it works
Romax Technology
Wind Energy
WEBDRUCK.CH
Web-To-Print Solution
Design and create individual printed matter
eShop with credit card payment
High quality PDF output with Printing Process integration
Thousands of orders for seven Swiss printing companies
Quuve
iBizLog - http://www.ibizlog.com
A product by Smallworks
SpesenFuchs
Pinesoft MBagger
Dedicated and cost-effective tools for software evolution
NOBIZZ-COM
BETTER
FASTER
CHEAPER
Yesplan is veelzijdige software voor het efficiënt plannen van evenementen.
NTed: disaster scenario recovery
airflowing
Organize your creative work
Plans and Pricing
OBJECT PROFILE
Das Content Management mit System
2denker
Continuous API Testing
keep your services under control 24/7
t3

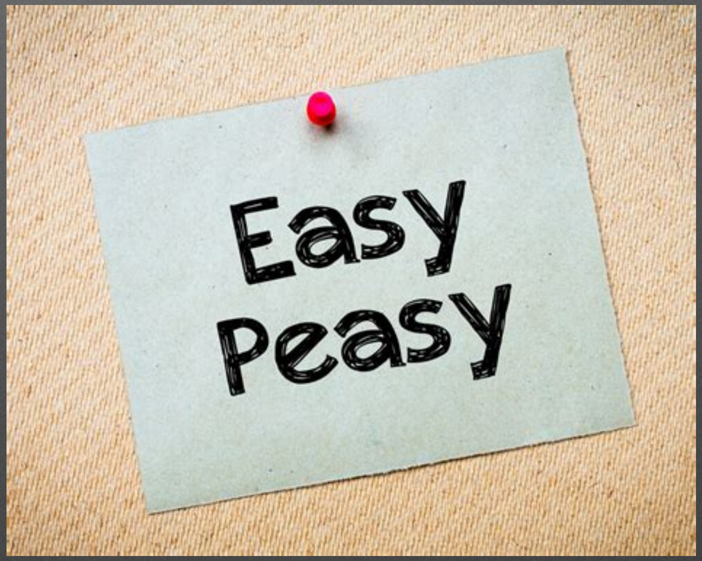
Some Success Stories

http://pharo.org/success

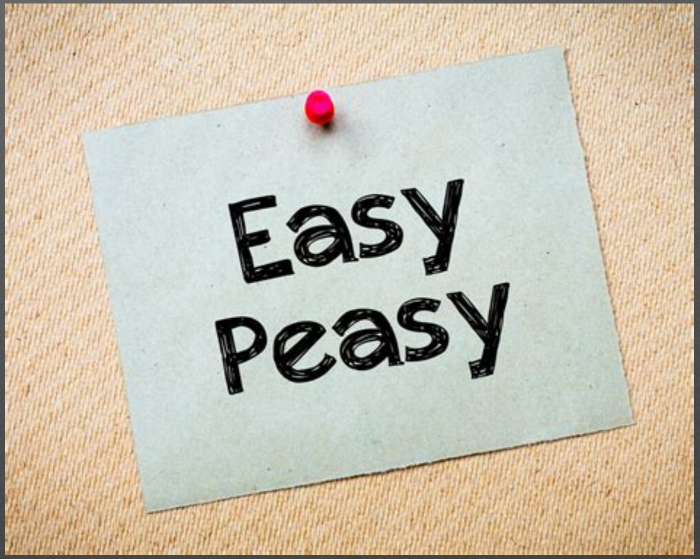
if you have a ‘success’ story
please send it to us!

Pharo?

Pharo?

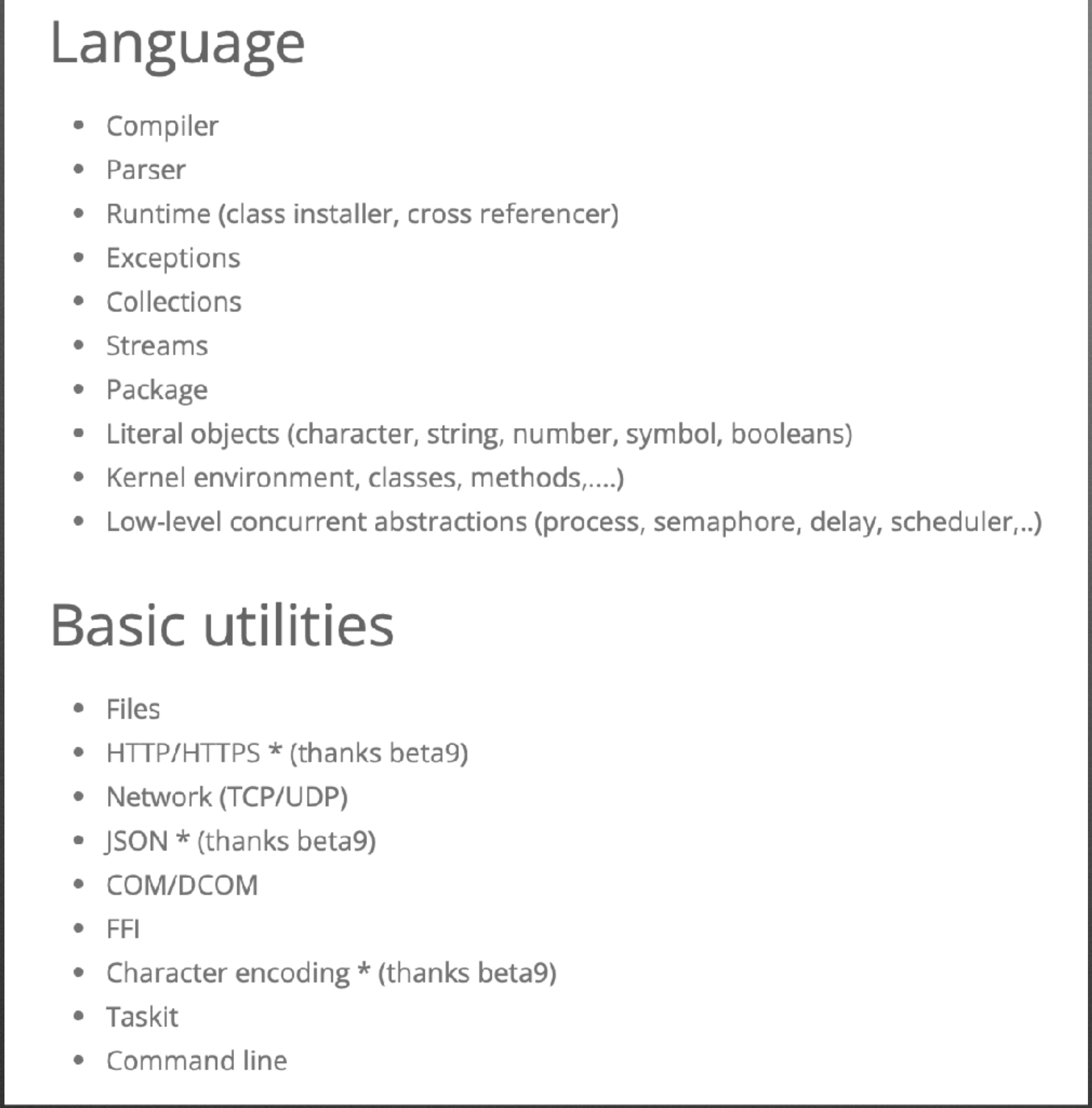


Pharo?



well…

https://pharo.org/aboutWhatIsPharo

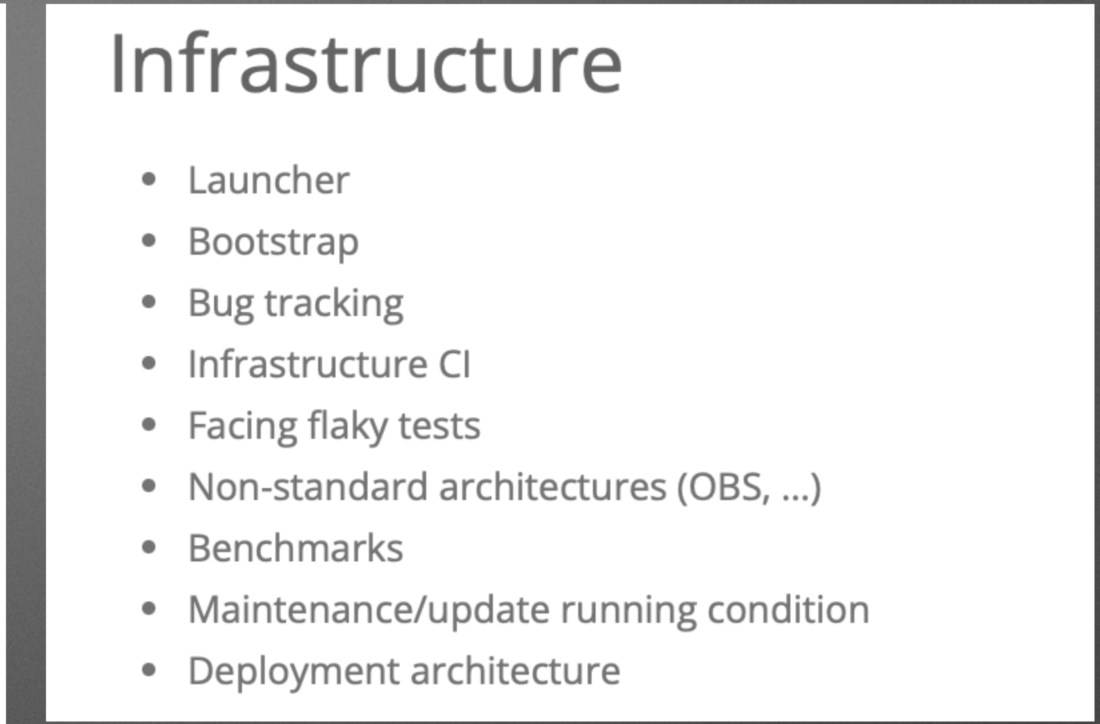


Language

- Compiler
- Parser
- Runtime (class installer, cross referencer)
- Exceptions
- Collections
- Streams
- Package
- Literal objects (character, string, number, symbol, booleans)
- Kernel environment, classes, methods,....)
- Low-level concurrent abstractions (process, semaphore, delay, scheduler,..)

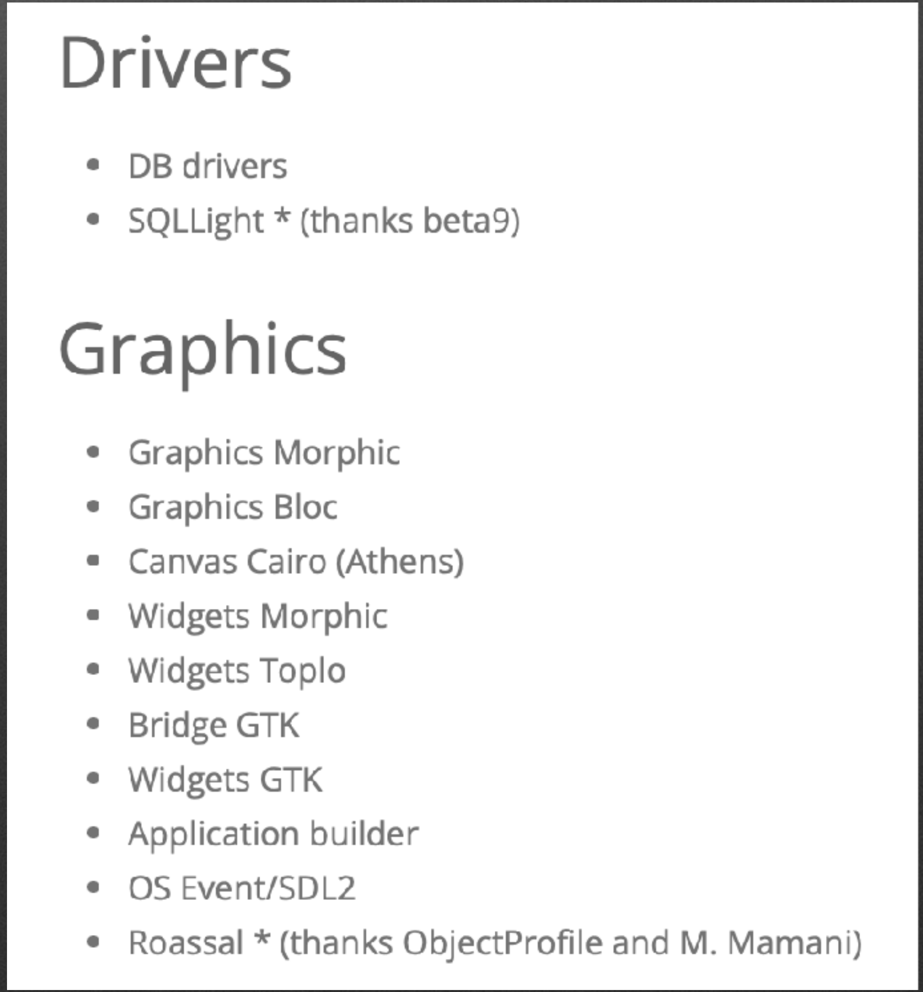
Basic utilities

- Files
- HTTP/HTTPS * (thanks beta9)
- Network (TCP/UDP)
- JSON * (thanks beta9)
- COM/DCOM
- FFI
- Character encoding * (thanks beta9)
- Taskit
- Command line



Infrastructure

- Launcher
- Bootstrap
- Bug tracking
- Infrastructure CI
- Facing flaky tests
- Non-standard architectures (OBS, ...)
- Benchmarks
- Maintenance/update running condition
- Deployment architecture



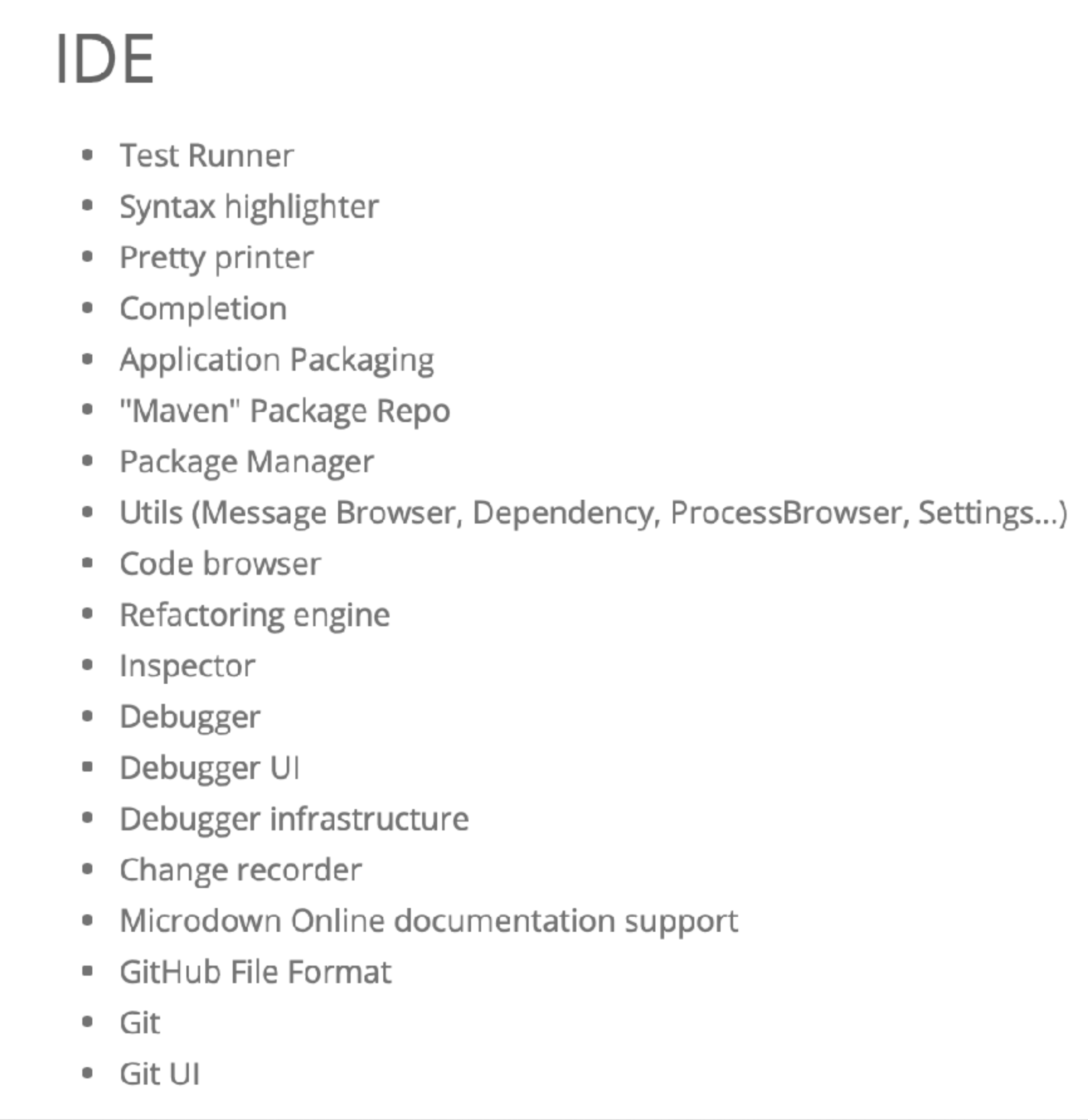
Drivers

- DB drivers
- SQLLight * (thanks beta9)

Graphics

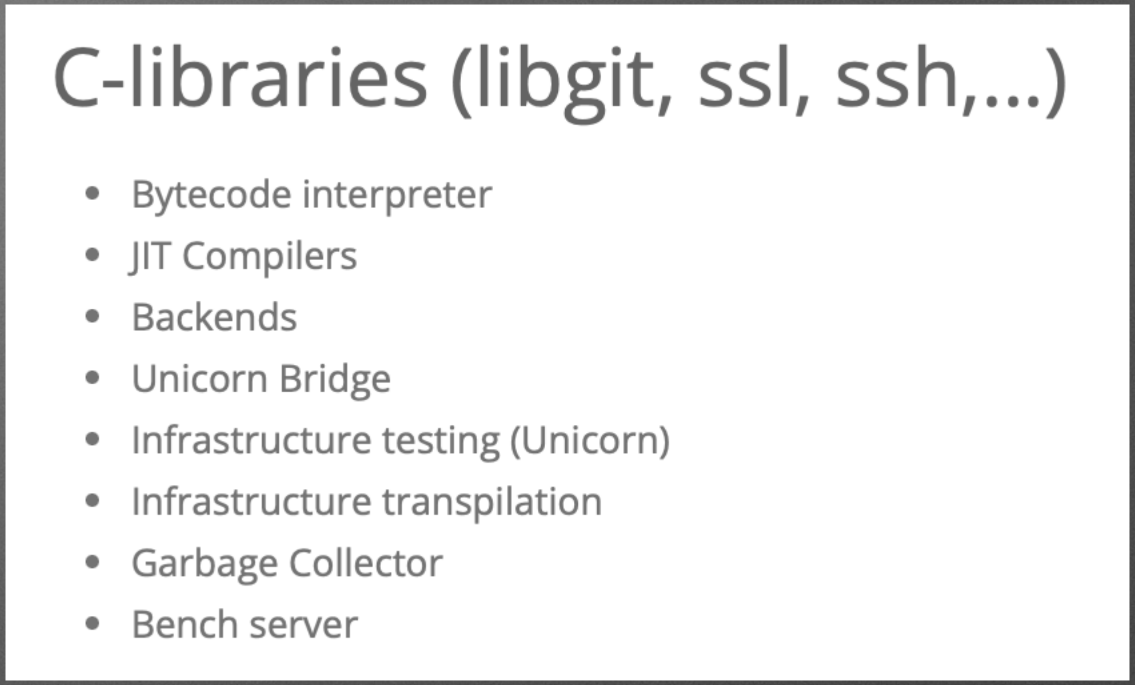
- Graphics Morphic
- Graphics Bloc
- Canvas Cairo (Athens)
- Widgets Morphic
- Widgets Toplo
- Bridge GTK
- Widgets GTK
- Application builder
- OS Event/SDL2
- Roassal * (thanks ObjectProfile and M. Mamani)

https://pharo.org/aboutWhatIsPharo



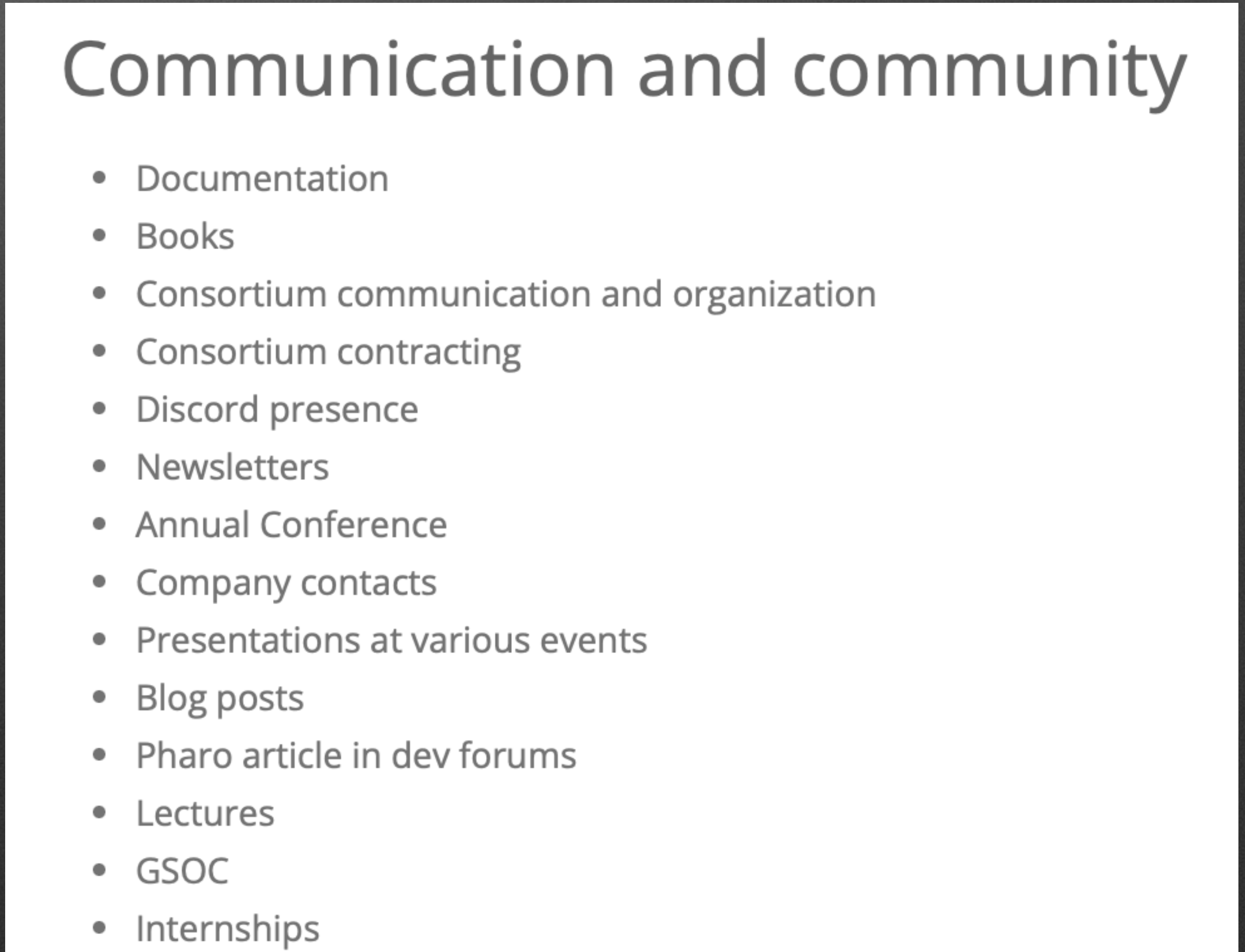
IDE

- Test Runner
- Syntax highlighter
- Pretty printer
- Completion
- Application Packaging
- "Maven" Package Repo
- Package Manager
- Utils (Message Browser, Dependency, ProcessBrowser, Settings...)
- Code browser
- Refactoring engine
- Inspector
- Debugger
- Debugger UI
- Debugger infrastructure
- Change recorder
- Microdown Online documentation support
- GitHub File Format
- Git
- Git UI



C-libraries (libgit, ssl, ssh,...)

- Bytecode interpreter
- JIT Compilers
- Backends
- Unicorn Bridge
- Infrastructure testing (Unicorn)
- Infrastructure transpilation
- Garbage Collector
- Bench server



Communication and community

- Documentation
- Books
- Consortium communication and organization
- Consortium contracting
- Discord presence
- Newsletters
- Annual Conference
- Company contacts
- Presentations at various events
- Blog posts
- Pharo article in dev forums
- Lectures
- GSOC
- Internships

A word about changes and support

Each time we change something we think in terms of impact and support

Each time we change
somet think in
terms ct and
support

(…)

We do maintain a LARGE code base and we do help people with old versions (recently we helped a company with Pharo 7.0)!

(…)

Yes Pharo7.0 and we are at Pharo12alpha…

Now economically we cannot do it all the times for free!

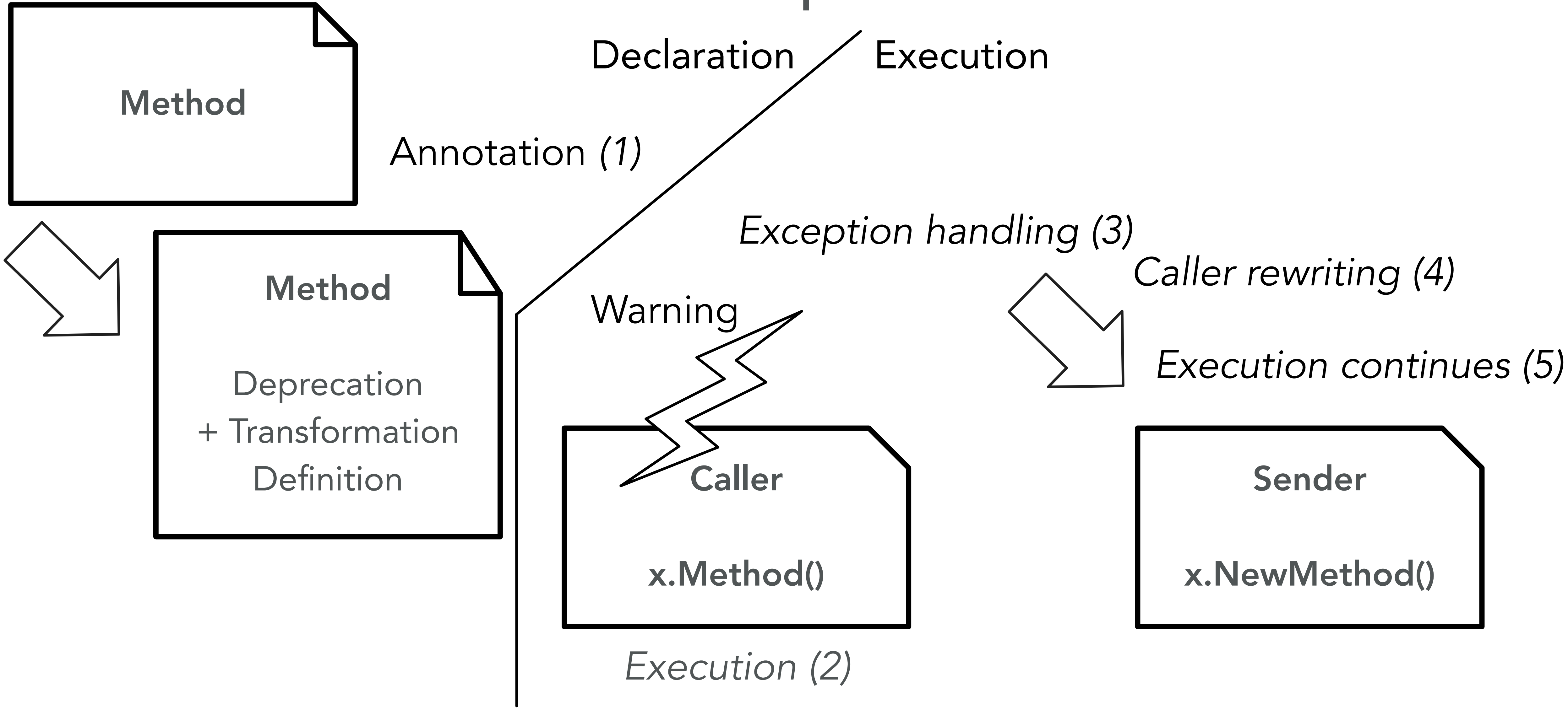
We do backports!

- [P11] Traits has to install method when is generated code #14155
  - https://github.com/pharo-project/pharo/pull/14155
- [P11] Update Spec #14446
  - https://github.com/pharo-project/pharo/pull/14446
- [P11] 14141-BlockClosureisClean-regression-from-Pharo-10-to-11 #14448
  - https://github.com/pharo-project/pharo/pull/14448
- [P11] SpMorphicBackendForTest>>#doubleClickFirstRowAndColumn: #1431
  - https://github.com/pharo-spec/Spec/pull/1431
- [P11] display scale factor improvements to Pharo 11 #1429
  - https://github.com/pharo-spec/Spec/pull/1429

Rewriting deprecations?

Truly unique

Deprewriter



Declaration

Execution

Method

Annotation *(1)*

Method

Deprecation
+ Transformation
Definition

Exception handling (3)

Caller rewriting (4)

Warning

Execution continues (5)

Caller

x.Method()

Sender

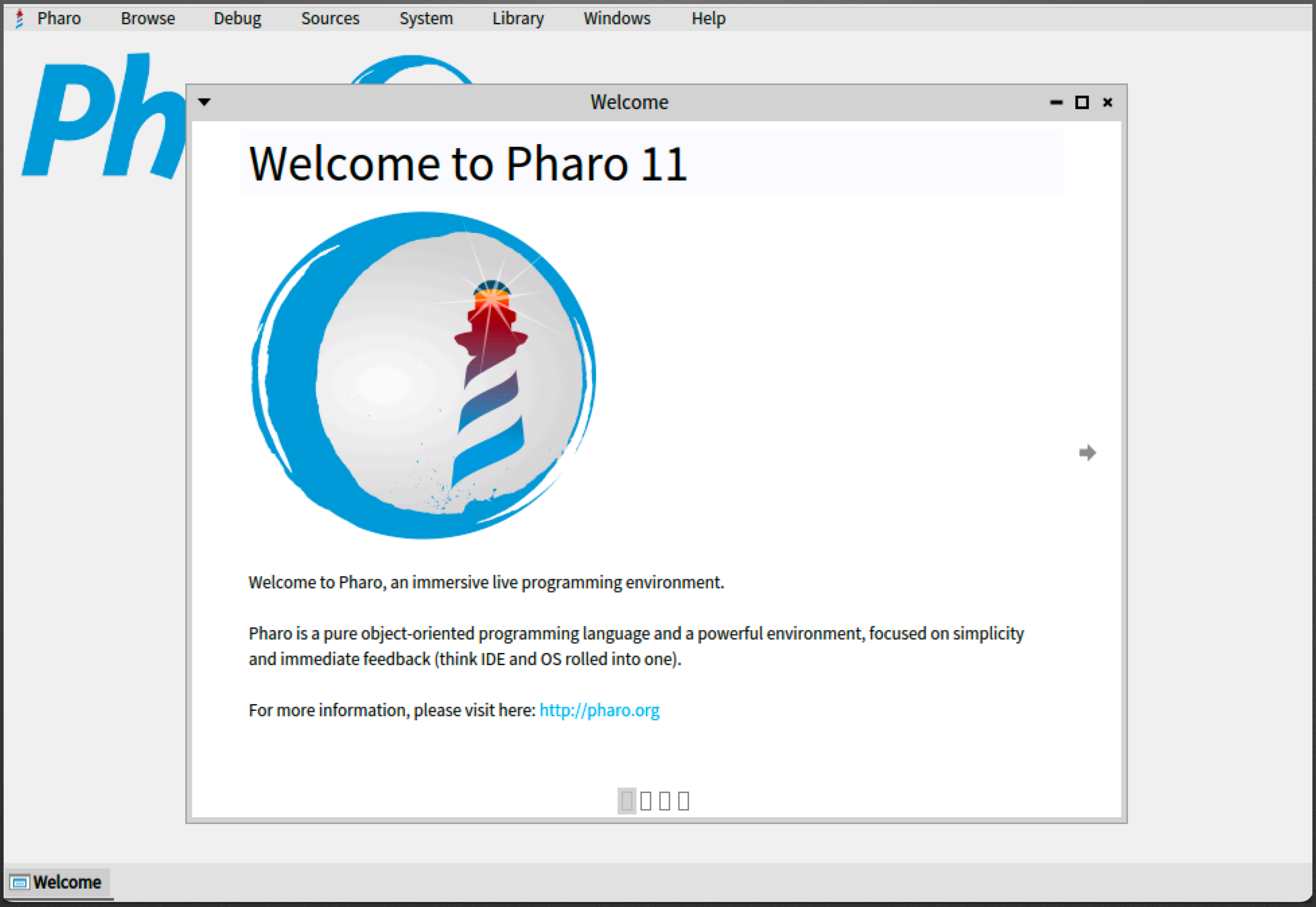
x.NewMethod()

Execution (2)

Rewriting deprecation

```
crLog: aString

    self
        deprecated: 'Please use trace* methods instead.'
        transformWith:
            '`@receiver crLog: `@statements1'
                -> '`@receiver crTrace: `@statements1'.
    self crTrace: aString
```

Run your tests.
Your code and your tests use the new API!



Welcome to Pharo 11

Welcome to Pharo, an immersive live programming environment.

Pharo is a pure object-oriented programming language and a powerful environment, focused on simplicity and immediate feedback (think IDE and OS rolled into one).

For more information, please visit here: http://pharo.org

Pharo 11: a stabilization iteration

- No big bang
- More fixes, more tests
- More documentation
- Better compiler
- Better VM
- Preparing next iteration

Large effort

- 1412 Pull Requests integrated *just* in the Pharo repository
- Closed 972 issues
- Contributions from more than 70 contributors

Pharo 11: Tools

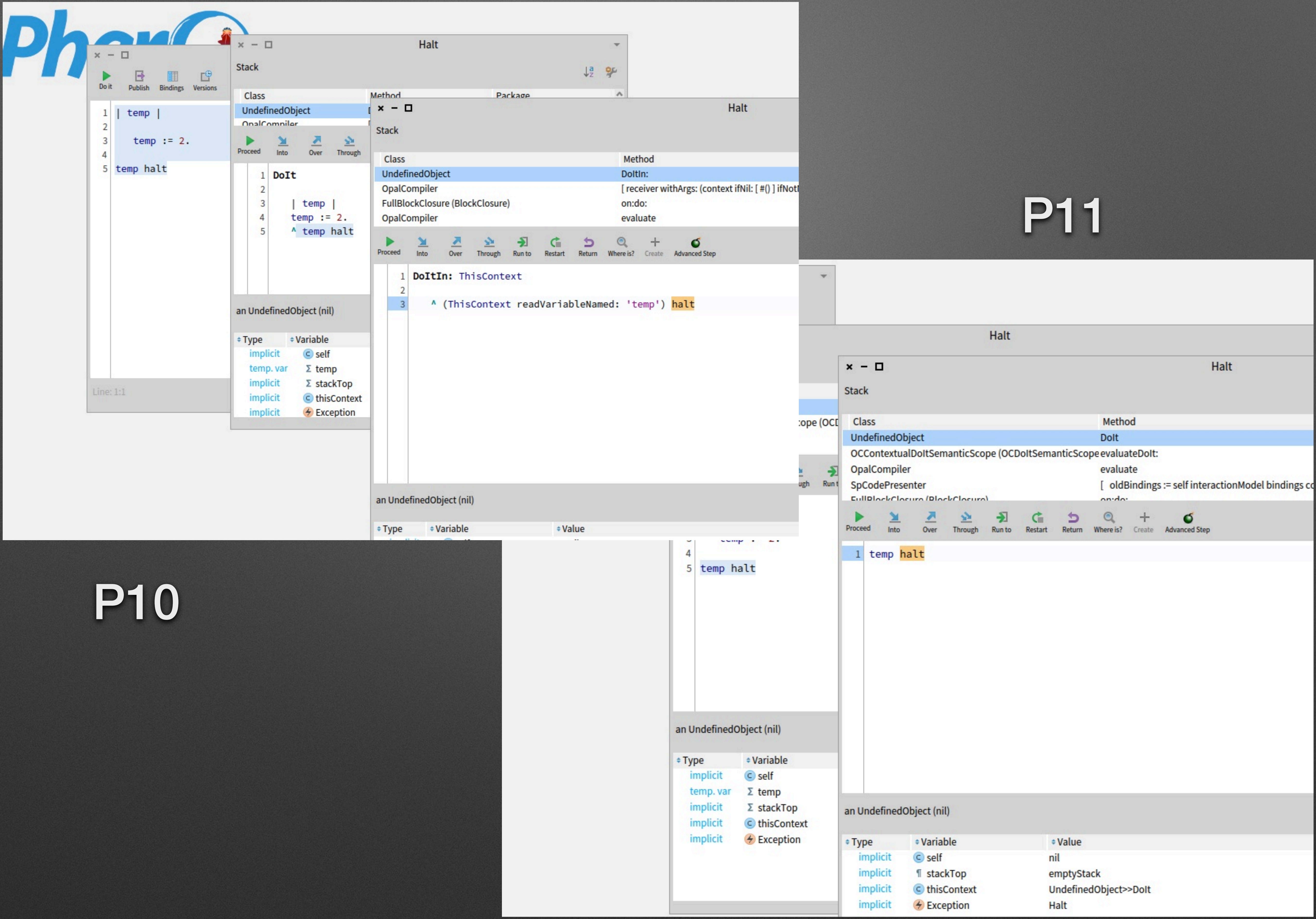
- Iceberg/Git fix and adaptation to github changes!
- Better debugger
- Finalization of adopting Fluid class definition
- Rewrite tools + better refactorings
- Better new tools
- DocumentationBrowser
- All versions of NewTools, Spec, Roassal and Microdown have been updated with their respective bug fixes and improvements

Systems

- Extended Full Block and Constant Block support
- Additional inlining and optimizations
- Bug fixes and clean up
- Ephemeron finalization support
- Permanent space

Compiler: Improved Doit

- No AST transformation
- No pretty printing
- No method header
- Take benefit of first class variables
- Looks more like what you wrote and not a hidden method



```
| temp |

  temp := 2.

temp halt
```

Class	Method	Package
UndefinedObject		
OpalCompiler		

```
DoIt

  | temp |
  temp := 2.
  ^ temp halt
```

an UndefinedObject (nil)

Type	Variable
implicit	self
temp. var	temp
implicit	stackTop
implicit	thisContext
implicit	Exception

Class	Method
UndefinedObject	DoItIn:
OpalCompiler	[receiver withArgs: (context ifNil: [#()] ifNot
FullBlockClosure (BlockClosure)	on:do:
OpalCompiler	evaluate

```
DoItIn: ThisContext

  ^ (ThisContext readVariableNamed: 'temp') halt
```

an UndefinedObject (nil)

Type	Variable	Value

P10

P11

```
4
5 temp halt
```

an UndefinedObject (nil)

Type	Variable
implicit	self
temp. var	temp
implicit	stackTop
implicit	thisContext
implicit	Exception

Class	Method
UndefinedObject	DoIt
OCContextualDoItSemanticScope (OCDoItSemanticScope	evaluateDoIt:
OpalCompiler	evaluate
SpCodePresenter	[oldBindings := self interactionModel bindings c
FullBlockClosure (BlockClosure)	on:do:

```
temp halt
```

an UndefinedObject (nil)

Type	Variable	Value
implicit	self	nil
implicit	stackTop	emptyStack
implicit	thisContext	UndefinedObject>>DoIt
implicit	Exception	Halt

Compiler: Improved Blocks

- Option: Full Blocks without outer context (if no return)
    - Faster, less memory use
    - Evaluating use by default
- Constant Block Closures are created at compile time

```
aDictionary at: #hello ifAbsentPut: [0]
```

Compiler: Optimizations

- optionInlineTimesRepeat and optionInlineRepeat are enabled by default

- No block evaluation e.g. for

```
1000 timesRepeat: [ self doSomething ]
```

Compiler: Misc

- Added a second plugin: “parse plugin” hook invoked after parsing
- Introduced new and improved Inspectors for AST/IR/Blocks

Compiler: looking ahead

- Huge cleaning started in P12
- Improving parsing logic
- Better handling of exceptions
- Another iteration of clean and constant blocks

- Thanks J. Privat

Pharo 11 VM

- Ephemerons Production Ready
- Permanent space + memory map (snapshot/startup)
- Initial support for Single-Instruction Multiple-Data (SIMD)
- Third-Party Dependency Update (Newer versions, Graphic Libraries using Hardware Acceleration)
- Clean Ups: Remove lots of old code, notably old experiments, and dead code

Pharo 11: VM

- Risc V JIT (ENSTA Bretagne)
- More tests
  - GC testing using smart fuzzers [ICST23]
  - Tests for interpreter/JIT equivalence [PLDI 2022]
- Slang improvements (GSOC and more)
- Revisit all the memory map (minimising swizzling)
  - VM start/snapshot

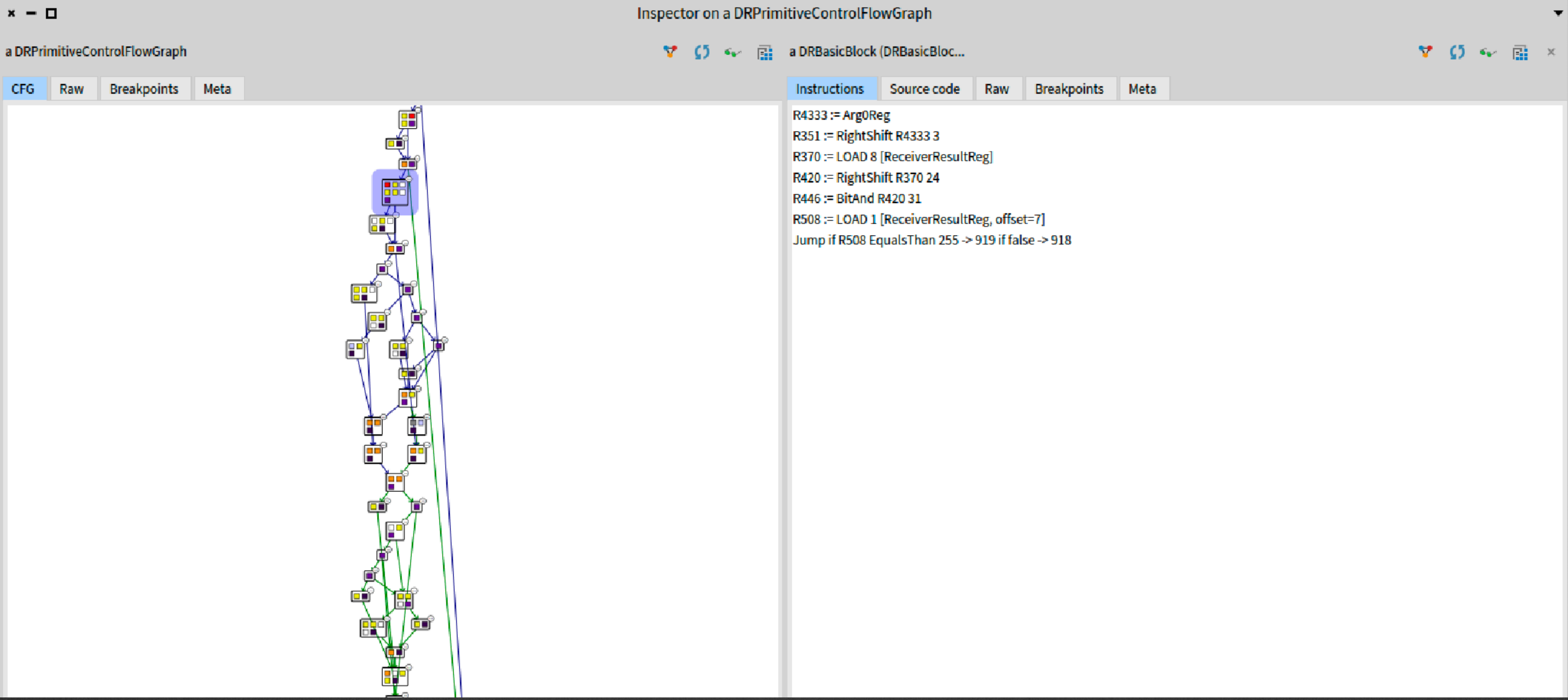
VM: Permanent Space

- Permanent space (sponsored by Lifeware)
- Memory region not GCed
- Permanent objects reduce GC pressure
- Manual choice
- Analysis tools to help you decide

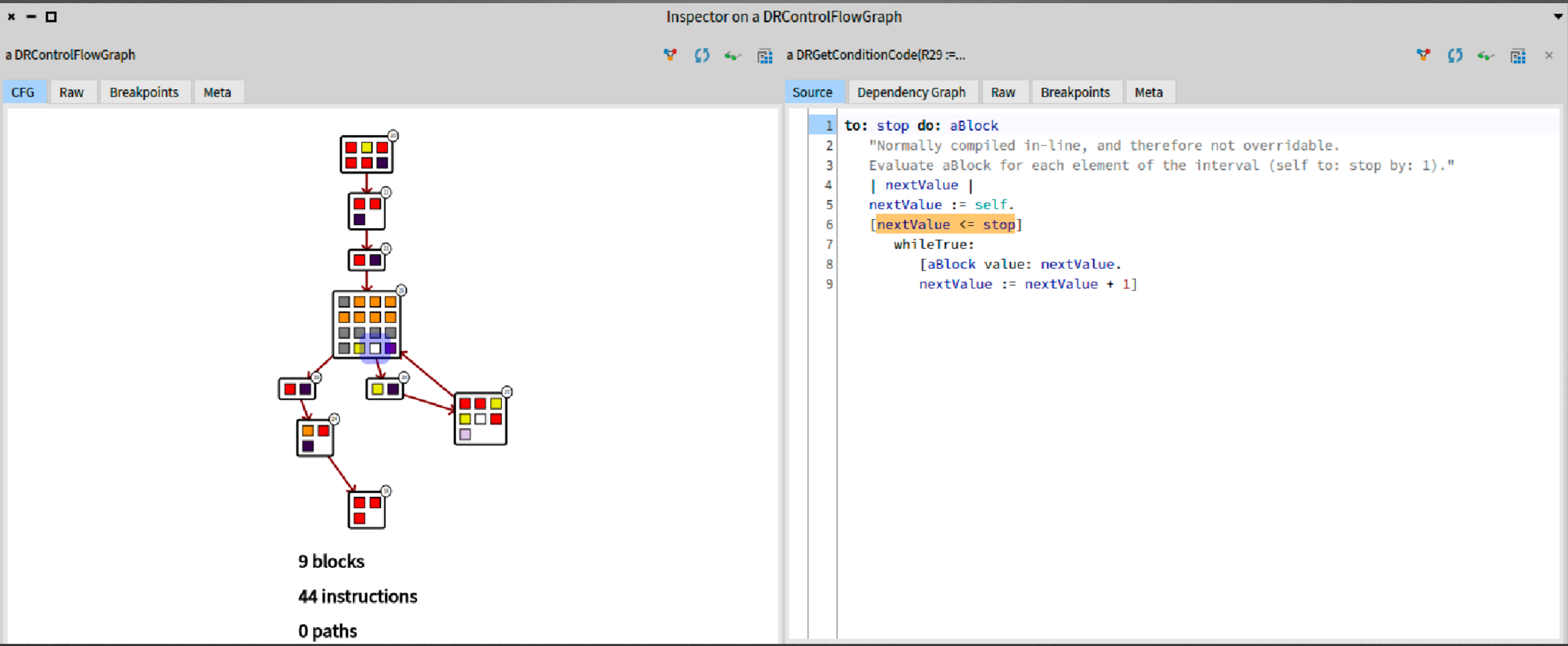
VM looking ahead

- More serious benchmarks
- Risc V JIT
- Druid: AOT (A head of time) compiler
  - Real compiler architecture Basic Blocks, CFG, Graph transformations
  - Nearly all byte-codes / primitives translated to machine code description language (to generate machine code at runtime)
- Dropal (Druid + Opal)
- Permanent space + memory mapping open a lot of new perspective (faster snapshot, shared memory region, segment,…)

AOT compiler CFG


Inspector on a DRPrimitiveControlFlowGraph
a DRPrimitiveControlFlowGraph
CFG
Raw
Breakpoints
Meta
a DRBasicBlock (DRBasicBloc...
Instructions
Source code
Raw
Breakpoints
Meta

```
R4333 := Arg0Reg
R351 := RightShift R4333 3
R370 := LOAD 8 [ReceiverResultReg]
R420 := RightShift R370 24
R446 := BitAnd R420 31
R508 := LOAD 1 [ReceiverResultReg, offset=7]
Jump if R508 EqualsThan 255 -> 919 if false -> 918
```



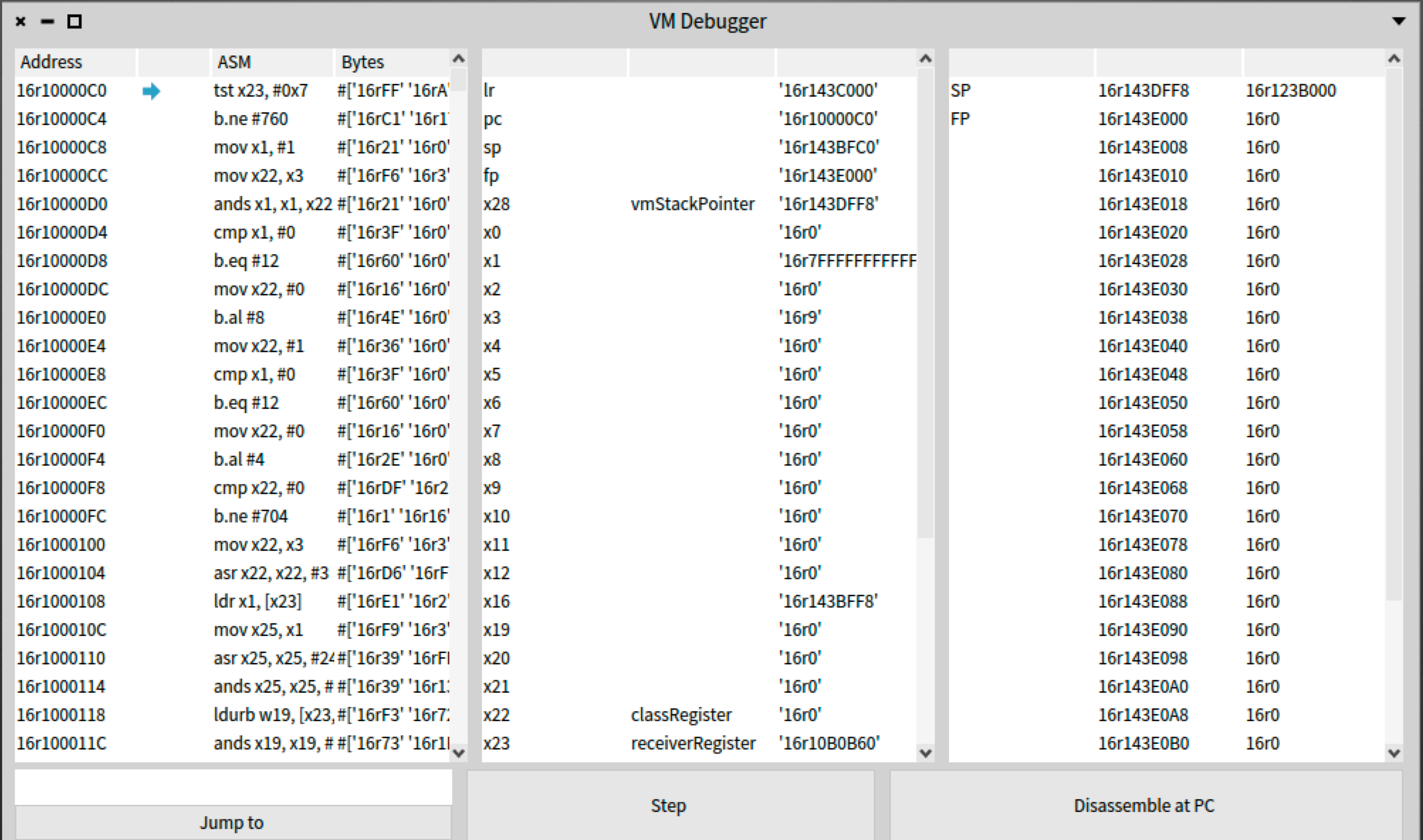
```
to: stop do: aBlock
   "Normally compiled in-line, and therefore not overridable.
   Evaluate aBlock for each element of the interval (self to: stop by: 1)."
   | nextValue |
   nextValue := self.
   [nextValue <= stop]
      whileTrue:
         [aBlock value: nextValue.
         nextValue := nextValue + 1]
```

9 blocks

44 instructions

0 paths

A little binary stepper



Address		ASM	Bytes
16r10000C0	→	tst x23, #0x7	#['16rFF' '16rA'
16r10000C4		b.ne #760	#['16rC1' '16r1
16r10000C8		mov x1, #1	#['16r21' '16r0'
16r10000CC		mov x22, x3	#['16rF6' '16r3'
16r10000D0		ands x1, x1, x22	#['16r21' '16r0'
16r10000D4		cmp x1, #0	#['16r3F' '16r0'
16r10000D8		b.eq #12	#['16r60' '16r0'
16r10000DC		mov x22, #0	#['16r16' '16r0'
16r10000E0		b.al #8	#['16r4E' '16r0'
16r10000E4		mov x22, #1	#['16r36' '16r0'
16r10000E8		cmp x1, #0	#['16r3F' '16r0'
16r10000EC		b.eq #12	#['16r60' '16r0'
16r10000F0		mov x22, #0	#['16r16' '16r0'
16r10000F4		b.al #4	#['16r2E' '16r0'
16r10000F8		cmp x22, #0	#['16rDF' '16r2
16r10000FC		b.ne #704	#['16r1' '16r16'
16r1000100		mov x22, x3	#['16rF6' '16r3'
16r1000104		asr x22, x22, #3	#['16rD6' '16rF
16r1000108		ldr x1, [x23]	#['16rE1' '16r2'
16r100010C		mov x25, x1	#['16rF9' '16r3'
16r1000110		asr x25, x25, #2	#['16r39' '16rF
16r1000114		ands x25, x25, #	#['16r39' '16r1
16r1000118		ldurb w19, [x23,	#['16rF3' '16r7
16r100011C		ands x19, x19, #	#['16r73' '16r1

lr		'16r143C000'
pc		'16r10000C0'
sp		'16r143BFC0'
fp		'16r143E000'
x28	vmStackPointer	'16r143DFF8'
x0		'16r0'
x1		'16r7FFFFFFFFFFF
x2		'16r0'
x3		'16r9'
x4		'16r0'
x5		'16r0'
x6		'16r0'
x7		'16r0'
x8		'16r0'
x9		'16r0'
x10		'16r0'
x11		'16r0'
x12		'16r0'
x16		'16r143BFF8'
x19		'16r0'
x20		'16r0'
x21		'16r0'
x22	classRegister	'16r0'
x23	receiverRegister	'16r10B0B60'

SP	16r143DFF8	16r123B000
FP	16r143E000	16r0
	16r143E008	16r0
	16r143E010	16r0
	16r143E018	16r0
	16r143E020	16r0
	16r143E028	16r0
	16r143E030	16r0
	16r143E038	16r0
	16r143E040	16r0
	16r143E048	16r0
	16r143E050	16r0
	16r143E058	16r0
	16r143E060	16r0
	16r143E068	16r0
	16r143E070	16r0
	16r143E078	16r0
	16r143E080	16r0
	16r143E088	16r0
	16r143E090	16r0
	16r143E098	16r0
	16r143E0A0	16r0
	16r143E0A8	16r0
	16r143E0B0	16r0

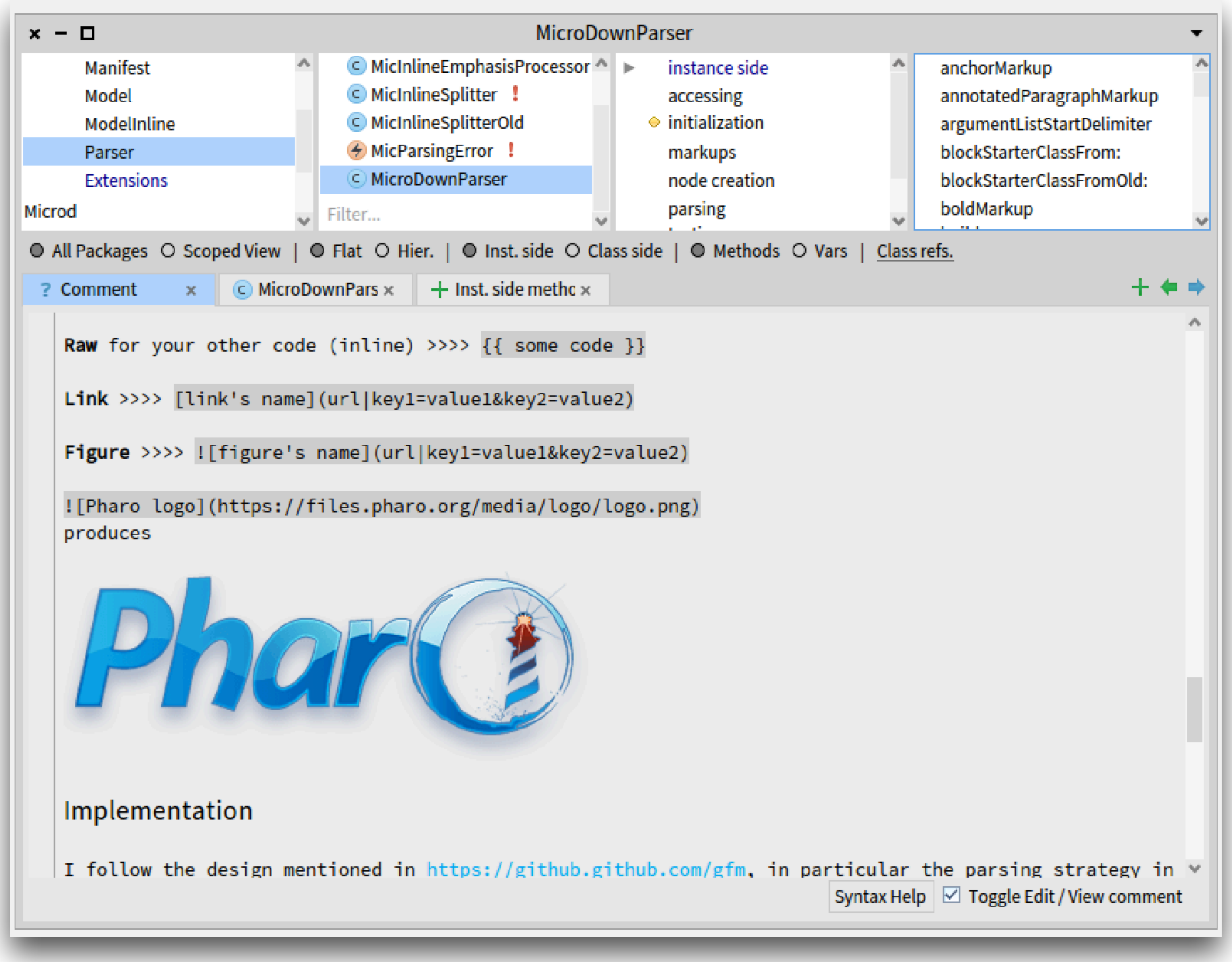
VM looking ahead

- Ready for a new iteration for more aggressive optimizations
- Strong understanding of the domain and how to get smart there
- Faster snapshots
  - Druid at runtime?
  - Support for static calls? (SIMD at your fingers)

Pharo 11: Documentation

- One format: microdown to rule them all
  - Github markdown ‘compliant’
  - Class comments / Class comment templates
  - Documentation
  - Books
- Big Thanks to K. Osterbye

Rendering of Class and Package Comments



Raw for your other code (inline) >>>> `{{ some code }}`

Link >>>> `[link's name](url|key1=value1&key2=value2)`

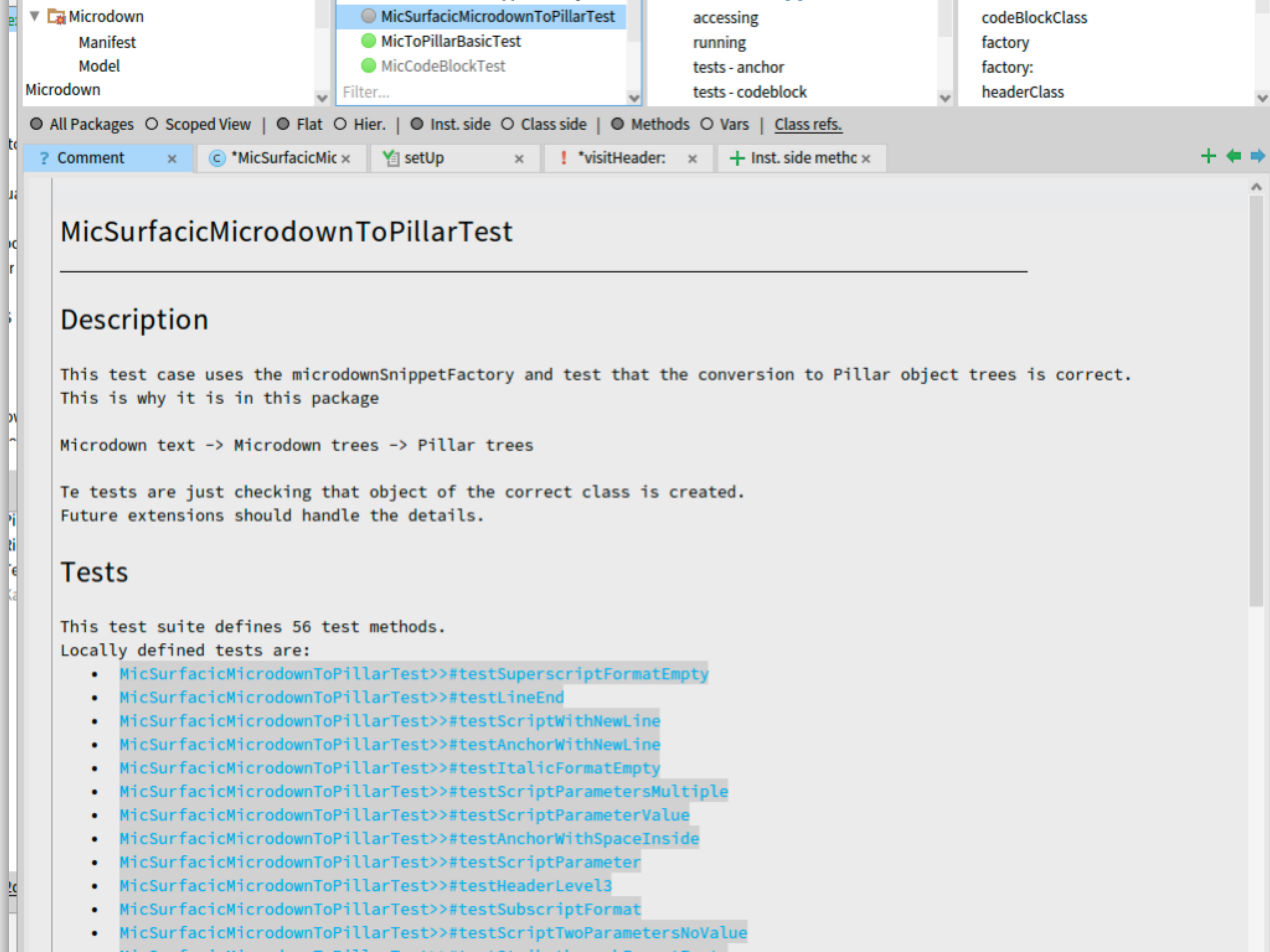
Figure >>>> `![figure's name](url|key1=value1&key2=value2)`

`![Pharo logo](https://files.pharo.org/media/logo/logo.png)`
produces

Pharo

Implementation

I follow the design mentioned in https://github.github.com/gfm, in particular the parsing strategy in



MicSurfacicMicrodownToPillarTest

Description

This test case uses the microdownSnippetFactory and test that the conversion to Pillar object trees is correct.
This is why it is in this package

Microdown text -> Microdown trees -> Pillar trees

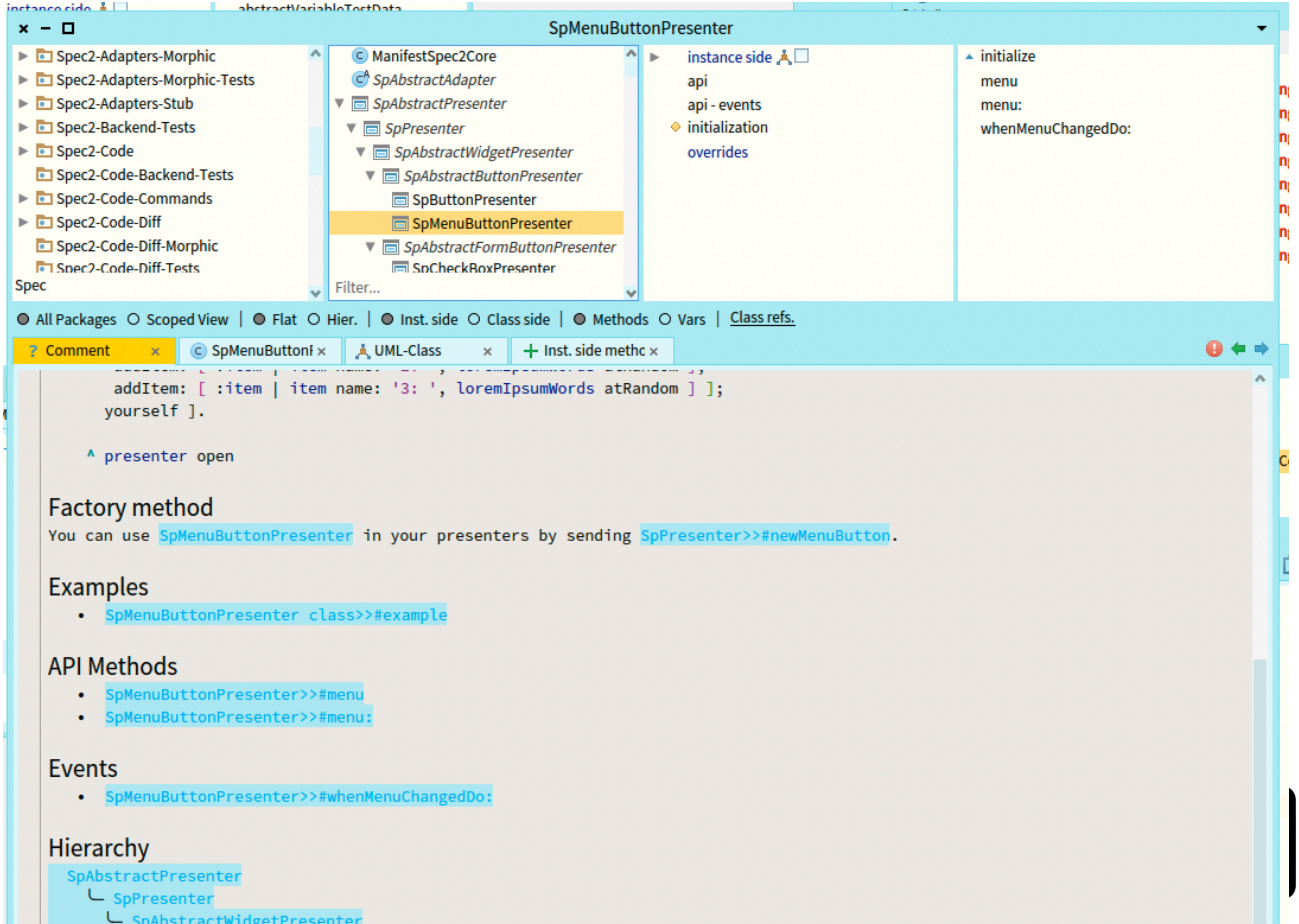
Te tests are just checking that object of the correct class is created.
Future extensions should handle the details.

Tests

This test suite defines 56 test methods.
Locally defined tests are:

- MicSurfacicMicrodownToPillarTest>>#testSuperscriptFormatEmpty
- MicSurfacicMicrodownToPillarTest>>#testLineEnd
- MicSurfacicMicrodownToPillarTest>>#testScriptWithNewLine
- MicSurfacicMicrodownToPillarTest>>#testAnchorWithNewLine
- MicSurfacicMicrodownToPillarTest>>#testItalicFormatEmpty
- MicSurfacicMicrodownToPillarTest>>#testScriptParametersMultiple
- MicSurfacicMicrodownToPillarTest>>#testScriptParameterValue
- MicSurfacicMicrodownToPillarTest>>#testAnchorWithSpaceInside
- MicSurfacicMicrodownToPillarTest>>#testScriptParameter
- MicSurfacicMicrodownToPillarTest>>#testHeaderLevel3
- MicSurfacicMicrodownToPillarTest>>#testSubscriptFormat
- MicSurfacicMicrodownToPillarTest>>#testScriptTwoParametersNoValue

Class Comment Templates



```
        addItem: [ :item | item name: '3: ', loremIpsumWords atRandom ] ];
       yourself ].

    ^ presenter open
```

Factory method

You can use SpMenuButtonPresenter in your presenters by sending SpPresenter>>#newMenuButton.

Examples

- SpMenuButtonPresenter class>>#example

API Methods

- SpMenuButtonPresenter>>#menu
- SpMenuButtonPresenter>>#menu:

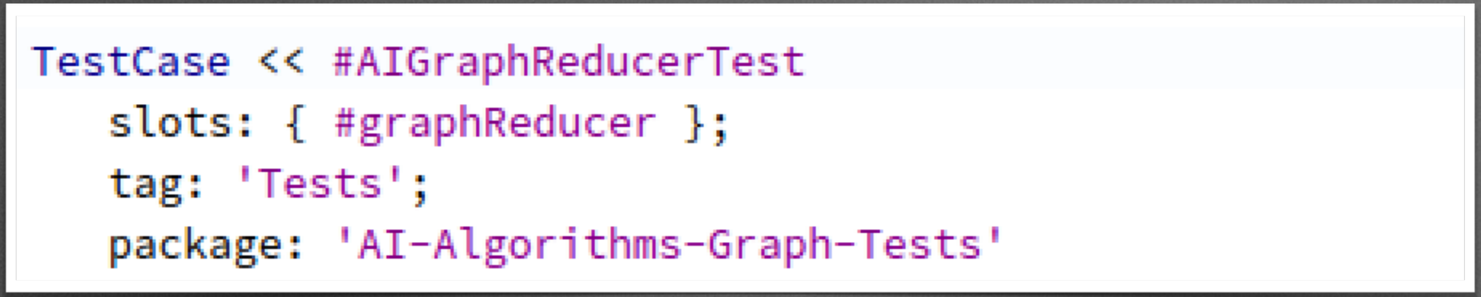
Events

- SpMenuButtonPresenter>>#whenMenuChangedDo:

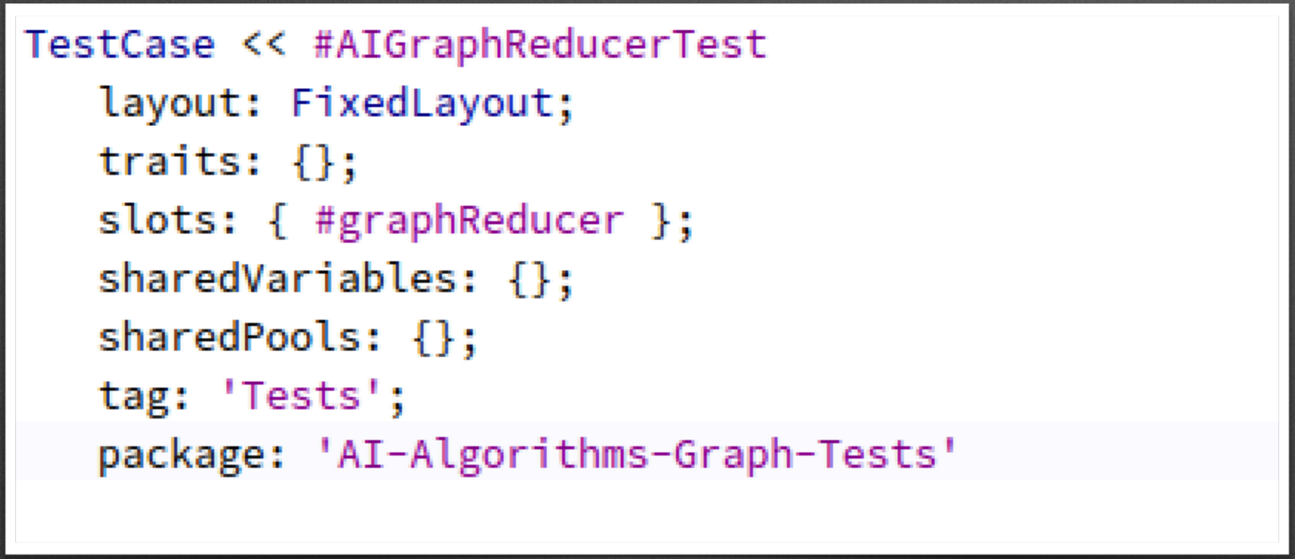
Hierarchy

```
SpAbstractPresenter
  └ SpPresenter
    └ SpAbstractWidgetPresenter
```

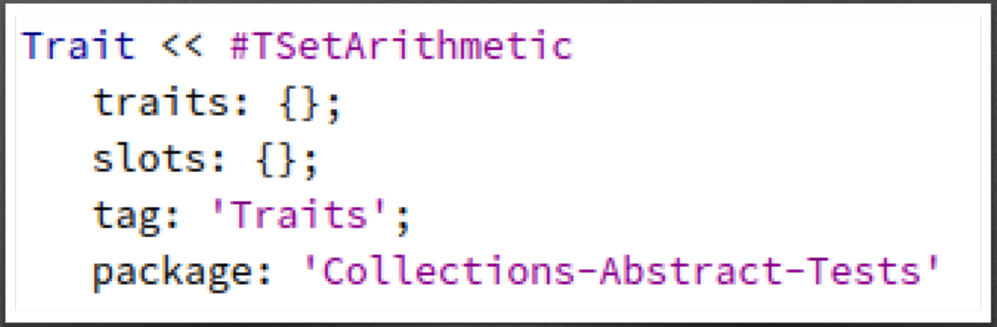
Fluid Class Syntax



```
TestCase << #AIGraphReducerTest
   slots: { #graphReducer };
   tag: 'Tests';
   package: 'AI-Algorithms-Graph-Tests'
```



```
TestCase << #AIGraphReducerTest
   layout: FixedLayout;
   traits: {};
   slots: { #graphReducer };
   sharedVariables: {};
   sharedPools: {};
   tag: 'Tests';
   package: 'AI-Algorithms-Graph-Tests'
```

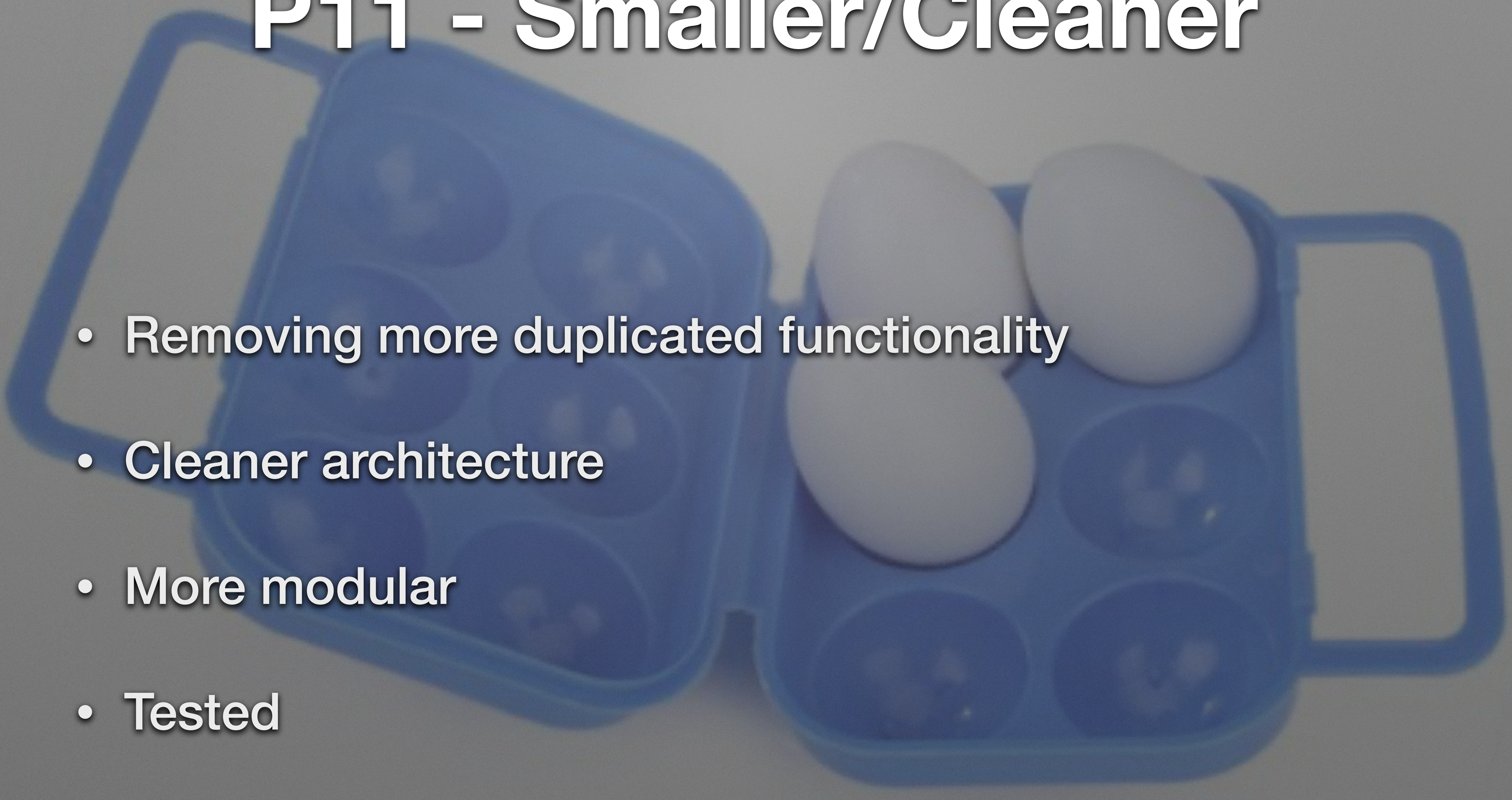


```
Trait << #TSetArithmetic
   traits: {};
   slots: {};
   tag: 'Traits';
   package: 'Collections-Abstract-Tests'
```

Fluid Class Syntax Trajectory

- Sketched and presented in 2017 at ESUG
- First release in P10 (took longer than we wanted)
    - Nice design
    - Scale well with multiple and optional parameters
    - Extensible
    - Clean and nice implementation
- P11: Default Pharo syntax!
- P12: Cleaning the left over

P11 - Smaller/Cleaner



- Removing more duplicated functionality
- Cleaner architecture
- More modular
- Tested

Spec20: a Cornerstone!

- Large reimplementation from Spec1 to Spec2
- Here to STAY!
- New widgets
- New layouts
- Multiple back ends (Morphic, GTK30, *Toplo*)

Testing Spec

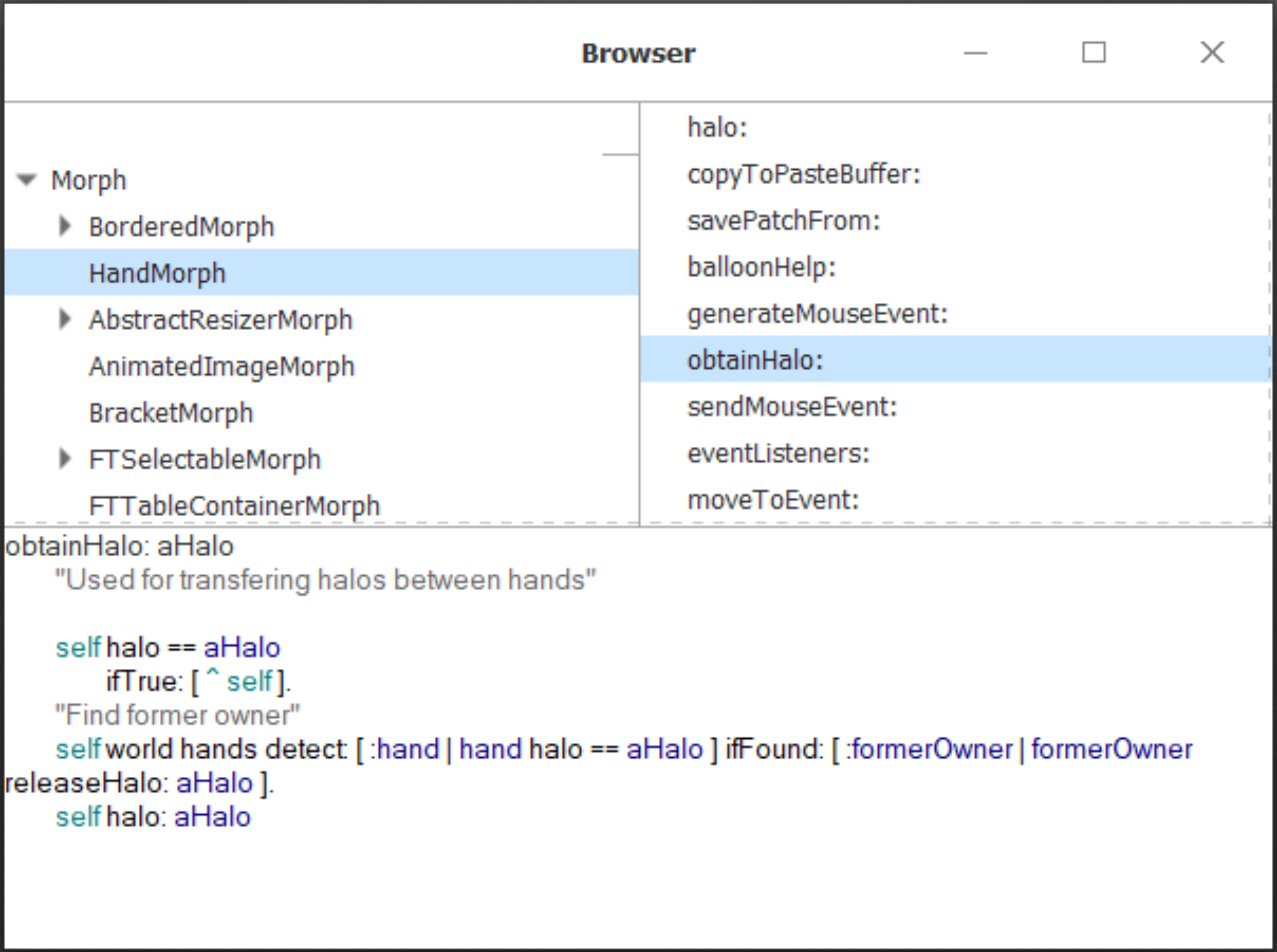
- Many tests
- Parameterized (same for Morphic and GTK30 and soon for TOPLO)

Presenters

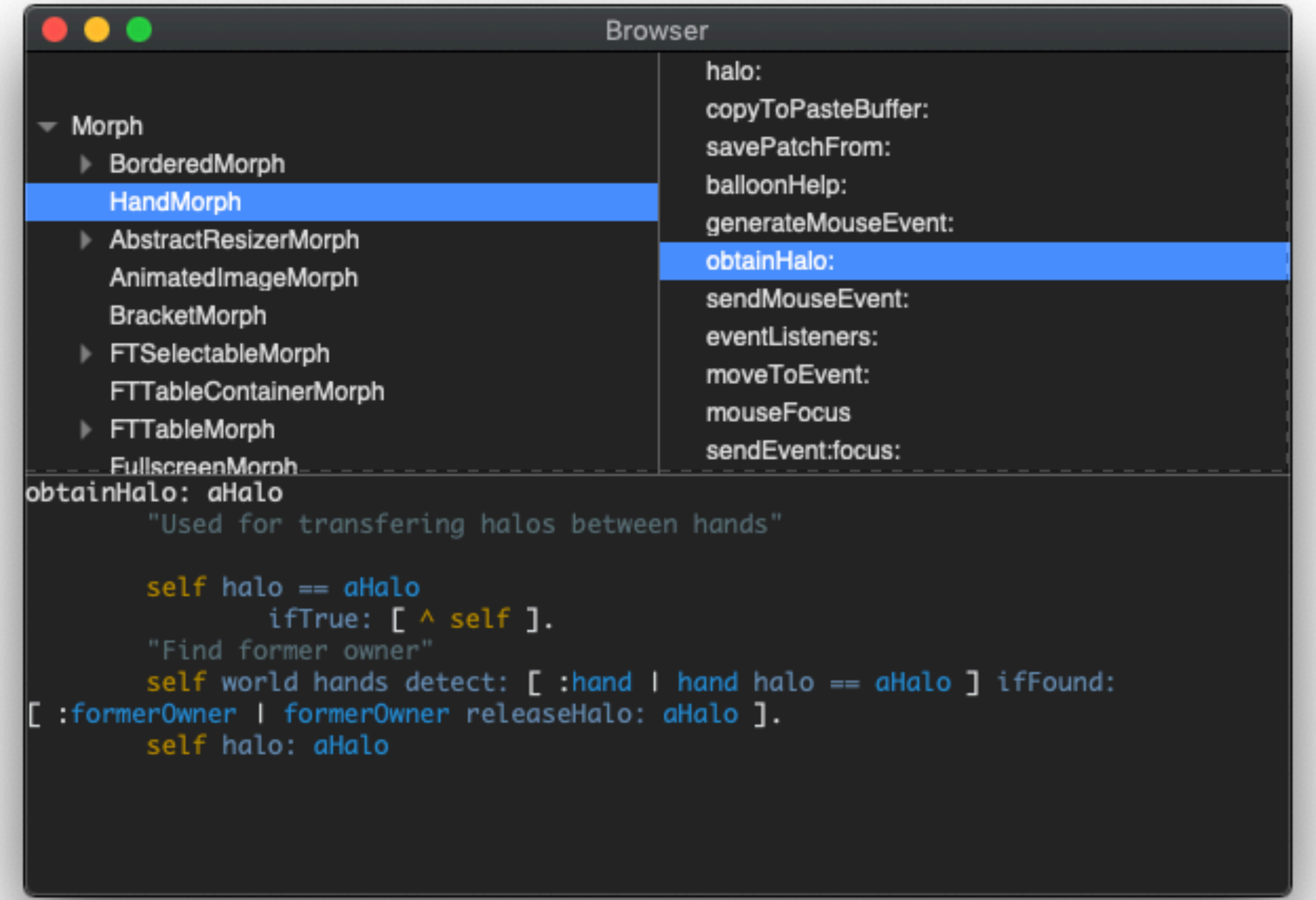
Layouts

Widgets

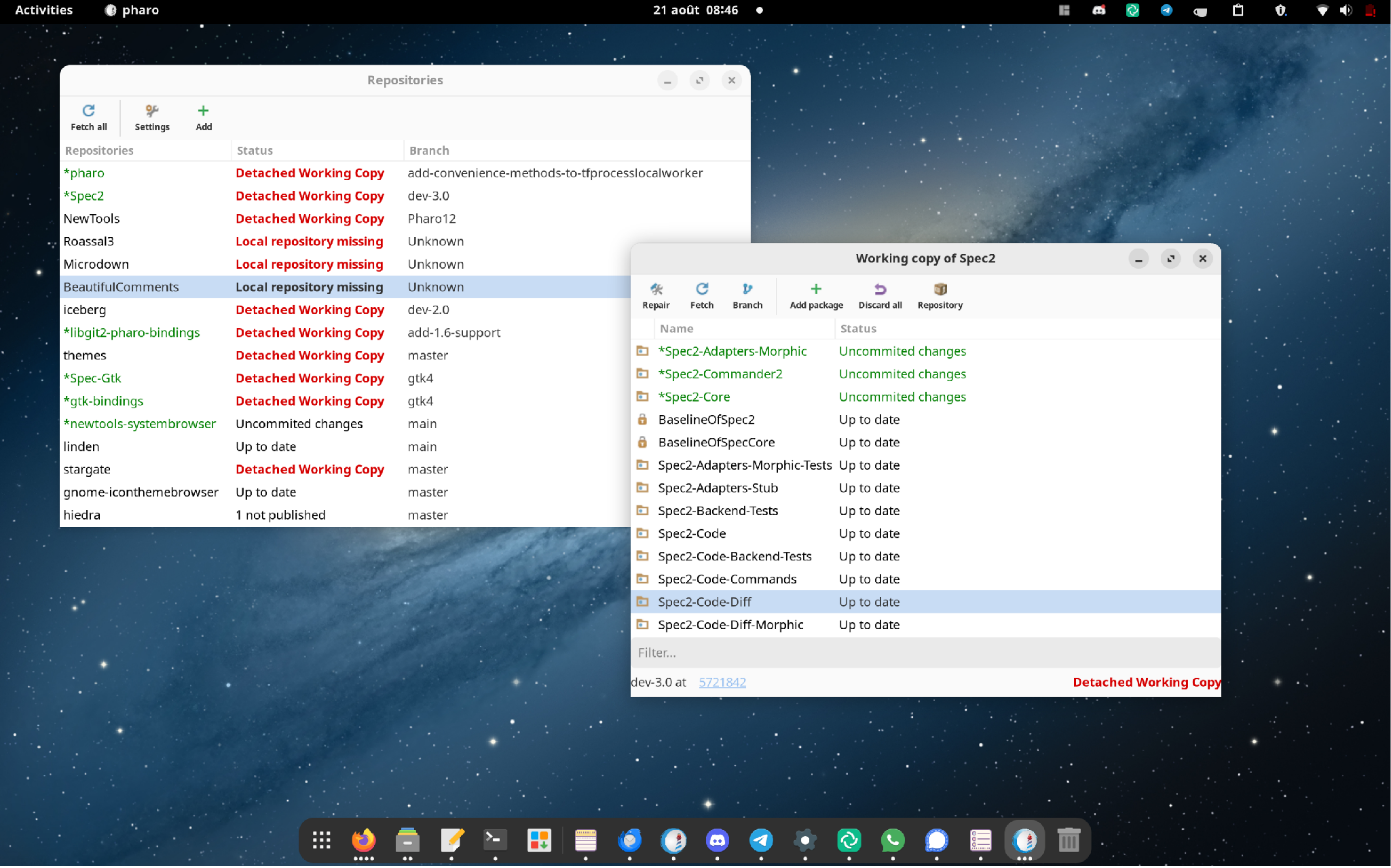
Adapters



```
obtainHalo: aHalo
    "Used for transfering halos between hands"

    self halo == aHalo
        ifTrue: [ ^ self ].
    "Find former owner"
    self world hands detect: [ :hand | hand halo == aHalo ] ifFound: [ :formerOwner | formerOwner
releaseHalo: aHalo ].
    self halo: aHalo
```



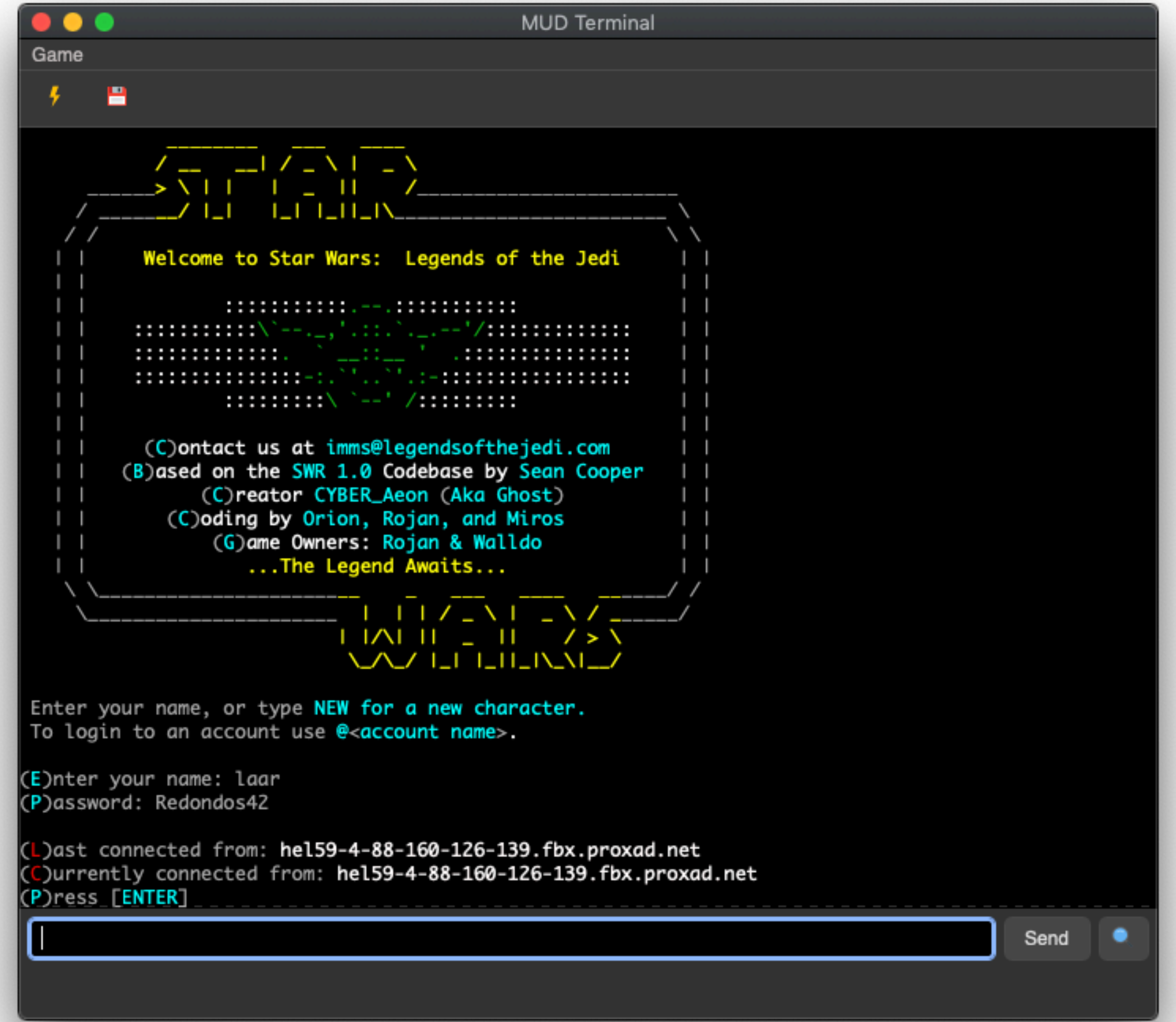
```
obtainHalo: aHalo
        "Used for transfering halos between hands"

        self halo == aHalo
                ifTrue: [ ^ self ].
        "Find former owner"
        self world hands detect: [ :hand | hand halo == aHalo ] ifFound:
[ :formerOwner | formerOwner releaseHalo: aHalo ].
        self halo: aHalo
```



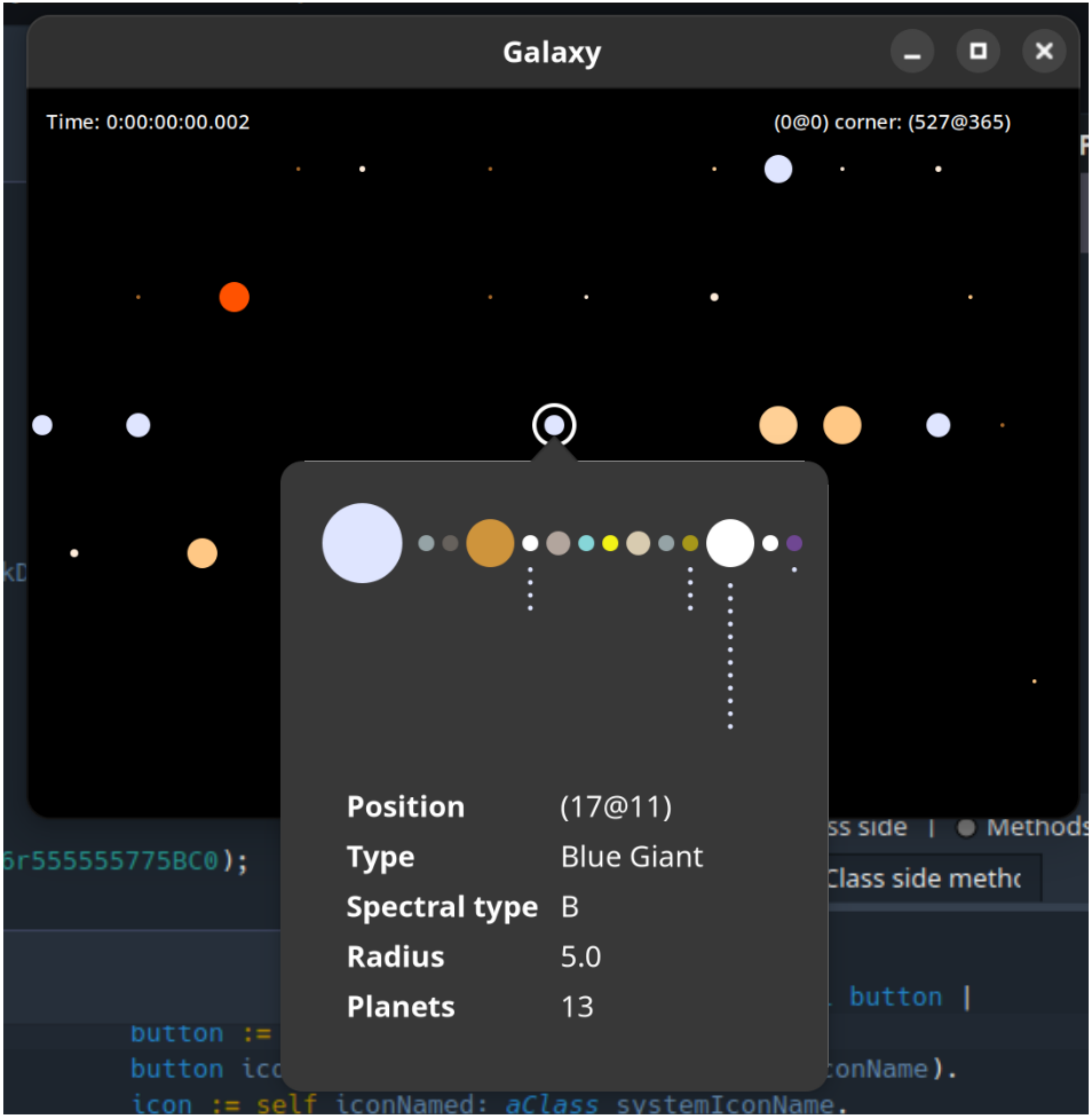
Repositories	Status	Branch
*pharo	Detached Working Copy	add-convenience-methods-to-tfprocesslocalworker
*Spec2	Detached Working Copy	dev-3.0
NewTools	Detached Working Copy	Pharo12
Roassal3	Local repository missing	Unknown
Microdown	Local repository missing	Unknown
BeautifulComments	Local repository missing	Unknown
iceberg	Detached Working Copy	dev-2.0
*libgit2-pharo-bindings	Detached Working Copy	add-1.6-support
themes	Detached Working Copy	master
*Spec-Gtk	Detached Working Copy	gtk4
*gtk-bindings	Detached Working Copy	gtk4
*newtools-systembrowser	Uncommited changes	main
linden	Up to date	main
stargate	Detached Working Copy	master
gnome-iconthemebrowser	Up to date	master
hiedra	1 not published	master

Name	Status
*Spec2-Adapters-Morphic	Uncommited changes
*Spec2-Commander2	Uncommited changes
*Spec2-Core	Uncommited changes
BaselineOfSpec2	Up to date
BaselineOfSpecCore	Up to date
Spec2-Adapters-Morphic-Tests	Up to date
Spec2-Adapters-Stub	Up to date
Spec2-Backend-Tests	Up to date
Spec2-Code	Up to date
Spec2-Code-Backend-Tests	Up to date
Spec2-Code-Commands	Up to date
Spec2-Code-Diff	Up to date
Spec2-Code-Diff-Morphic	Up to date

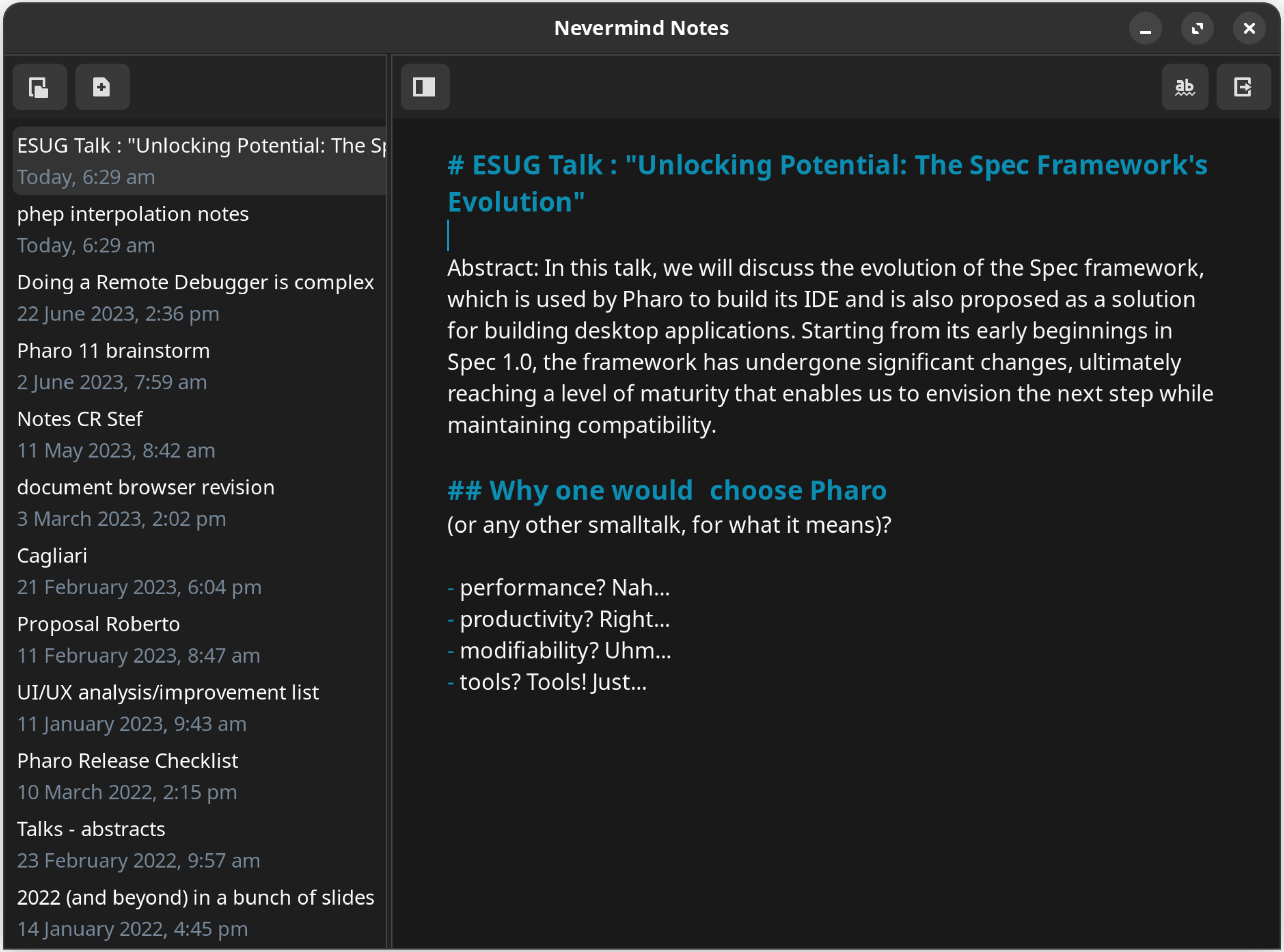
dev-3.0 at 5721842 Detached Working Copy



```
             ________   ___   ____
            / __   __| / _ \ |  _ \
      ______> \ | |   |  _  ||    /_______________________
     / _______/ |_|   |_| |_||_|\_________________________ \
    / /                                                   \ \
   | |     Welcome to Star Wars:  Legends of the Jedi      | |
   | |                                                     | |
   | |            :::::::::::.--.:::::::::::               | |
   | |    :::::::::::\`--._,'.::.`._.--'/:::::::::::::     | |
   | |    :::::::::::::.  ` __::__ '  .:::::::::::::::     | |
   | |    :::::::::::::::-:.`'..`'.:-:::::::::::::::::     | |
   | |            :::::::::\ `--' /:::::::::               | |
   | |                                                     | |
   | |     (C)ontact us at imms@legendsofthejedi.com       | |
   | |   (B)ased on the SWR 1.0 Codebase by Sean Cooper    | |
   | |          (C)reator CYBER_Aeon (Aka Ghost)           | |
   | |       (C)oding by Orion, Rojan, and Miros           | |
   | |           (G)ame Owners: Rojan & Walldo             | |
   | |              ...The Legend Awaits...                | |
    \ \_______________________    _   ___   ____   _______/ /
     \______________________  |  | | / _ \ |  _ \ / _______/
                            | |/\| ||  _  ||    / > \
                             \_/\_/ |_| |_||_|\_\|__/

 Enter your name, or type NEW for a new character.
 To login to an account use @<account name>.

(E)nter your name: laar
(P)assword: Redondos42

(L)ast connected from: hel59-4-88-160-126-139.fbx.proxad.net
(C)urrently connected from: hel59-4-88-160-126-139.fbx.proxad.net
(P)ress [ENTER]
```



Position	(17@11)
Type	Blue Giant
Spectral type	B
Radius	5.0
Planets	13



ESUG Talk : "Unlocking Potential: The Sp
Today, 6:29 am

phep interpolation notes
Today, 6:29 am

Doing a Remote Debugger is complex
22 June 2023, 2:36 pm

Pharo 11 brainstorm
2 June 2023, 7:59 am

Notes CR Stef
11 May 2023, 8:42 am

document browser revision
3 March 2023, 2:02 pm

Cagliari
21 February 2023, 6:04 pm

Proposal Roberto
11 February 2023, 8:47 am

UI/UX analysis/improvement list
11 January 2023, 9:43 am

Pharo Release Checklist
10 March 2022, 2:15 pm

Talks - abstracts
23 February 2022, 9:57 am

2022 (and beyond) in a bunch of slides
14 January 2022, 4:45 pm

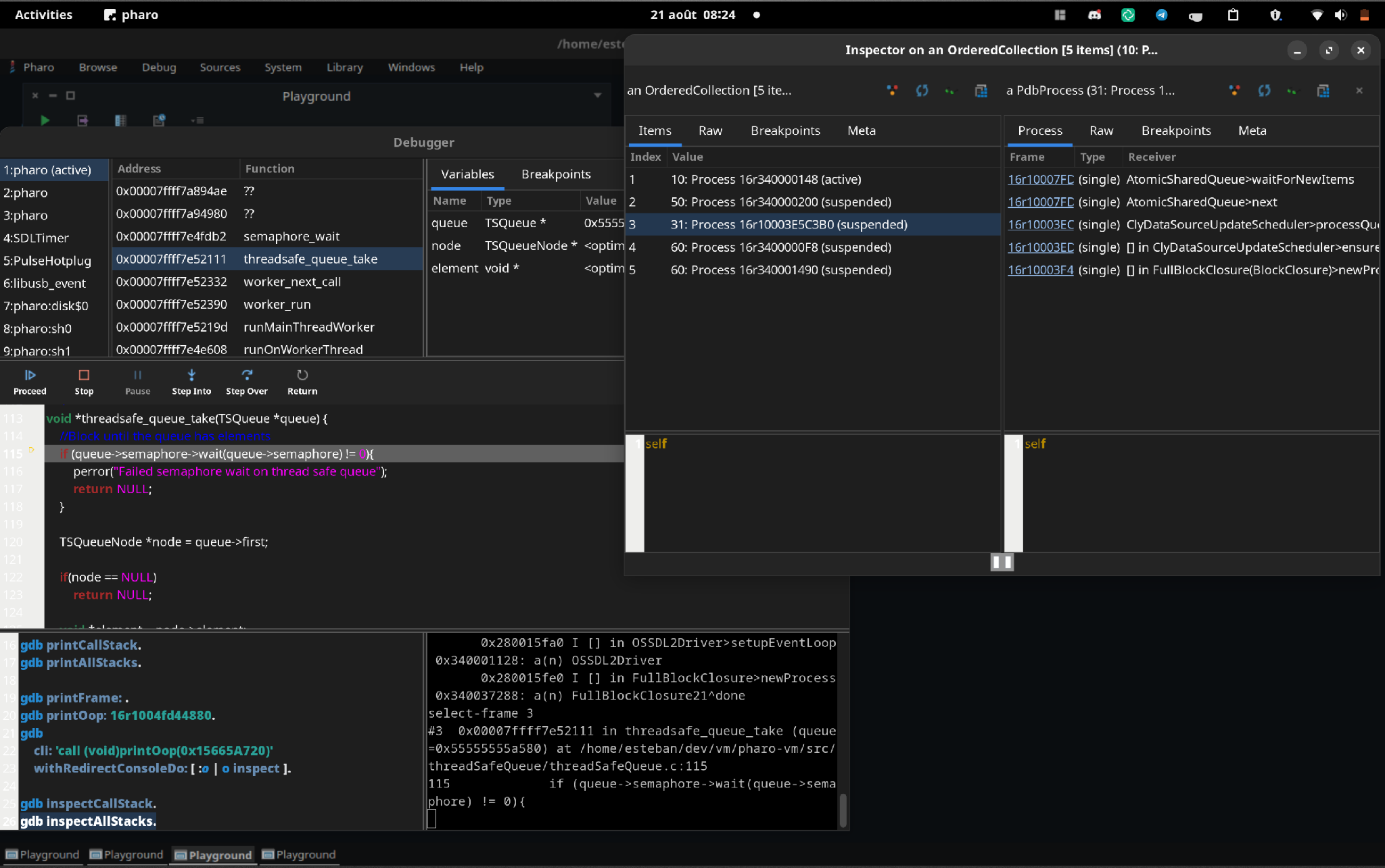
ESUG Talk : "Unlocking Potential: The Spec Framework's Evolution"

Abstract: In this talk, we will discuss the evolution of the Spec framework, which is used by Pharo to build its IDE and is also proposed as a solution for building desktop applications. Starting from its early beginnings in Spec 1.0, the framework has undergone significant changes, ultimately reaching a level of maturity that enables us to envision the next step while maintaining compatibility.

Why one would choose Pharo

(or any other smalltalk, for what it means)?

- performance? Nah...
- productivity? Right...
- modifiability? Uhm...
- tools? Tools! Just...



	Address	Function
1:pharo (active)	0x00007ffff7a894ae	??
2:pharo	0x00007ffff7a94980	??
3:pharo	0x00007ffff7e4fdb2	semaphore_wait
4:SDLTimer	0x00007ffff7e52111	threadsafe_queue_take
5:PulseHotplug	0x00007ffff7e52332	worker_next_call
6:libusb_event	0x00007ffff7e52390	worker_run
7:pharo:disk$0	0x00007ffff7e5219d	runMainThreadWorker
8:pharo:sh0	0x00007ffff7e4e608	runOnWorkerThread
9:pharo:sh1		

Name	Type	Value
queue	TSQueue *	0x5555
node	TSQueueNode *	<optim
element	void *	<optim

```
void *threadsafe_queue_take(TSQueue *queue) {
  //Block until the queue has elements
  if (queue->semaphore->wait(queue->semaphore) != 0){
    perror("Failed semaphore wait on thread safe queue");
    return NULL;
  }

  TSQueueNode *node = queue->first;

  if(node == NULL)
    return NULL;
```

Index	Value
1	10: Process 16r340000148 (active)
2	50: Process 16r340000200 (suspended)
3	31: Process 16r10003E5C3B0 (suspended)
4	60: Process 16r3400000F8 (suspended)
5	60: Process 16r340001490 (suspended)

Frame	Type	Receiver
16r10007FC	(single)	AtomicSharedQueue>waitForNewItems
16r10007FC	(single)	AtomicSharedQueue>next
16r10003EC	(single)	ClyDataSourceUpdateScheduler>processQu
16r10003EC	(single)	[] in ClyDataSourceUpdateScheduler>ensure
16r10003F4	(single)	[] in FullBlockClosure(BlockClosure)>newPro

```
gdb printCallStack.
gdb printAllStacks.

gdb printFrame: .
gdb printOop: 16r1004fd44880.
gdb
  cli: 'call (void)printOop(0x15665A720)'
  withRedirectConsoleDo: [ :o | o inspect ].

gdb inspectCallStack.
gdb inspectAllStacks.
```

```
          0x280015fa0 I [] in OSSDL2Driver>setupEventLoop
 0x340001128: a(n) OSSDL2Driver
          0x280015fe0 I [] in FullBlockClosure>newProcess
 0x340037288: a(n) FullBlockClosure21^done
select-frame 3
#3  0x00007ffff7e52111 in threadsafe_queue_take (queue
=0x55555555a580) at /home/esteban/dev/vm/pharo-vm/src/
threadSafeQueue/threadSafeQueue.c:115
115             if (queue->semaphore->wait(queue->sema
phore) != 0){
```

The trajectory

- Converting all existing tools to use Spec
- Support mix of back-ends
- Have backends (GTK, Morphic, Toplo) for Spec
- Remove Morphic and use Bloc/Toplo

Spec20 in P11

- Maturation phase (no breaking changes)
  - Fixed problems on layout behavior, particularly on SpBoxLayout.
  - Enhanced the way styles work (on Morphic).
  - General presenters improvement and add some common usage widgets.
- Overall, ~80 issues processed.

Layout fixes

- All layouts received a pass to make them more adaptable
- SpBoxLayout and SpScrollableLayout added generic align properties (vAlignStart, vAlignCenter, vAlignEnd, hAlignStart, hAlignCenter, hAlignEnd)

Style enhancements in Morphic

- Morphic is not well prepared to be styled, we added a lot of hooks to make it possible where it was not before (like in buttons)

- They are now stateful part of the configuration (and can be reset to see changes)

- They can now react to theme changes (from dark to light), and in morphic they can use theme color palette.

- They can now scale the components when you scale the world

Misc: Presenter improvements

- Add context menus to several presenters (SpMorphPresenter and others)

- Tables can have alternating row colors

- Added common widgets to be reused: SpChooserPresenter, SpFilteringListPresenter...

- New standard dialogs using a builder pattern (adds more control on behavior)

Other presenters

- Roassal presenter
- Microdown presenter
- GTK specific presenters: Vte, WebKitGtk

Spec: Looking ahead

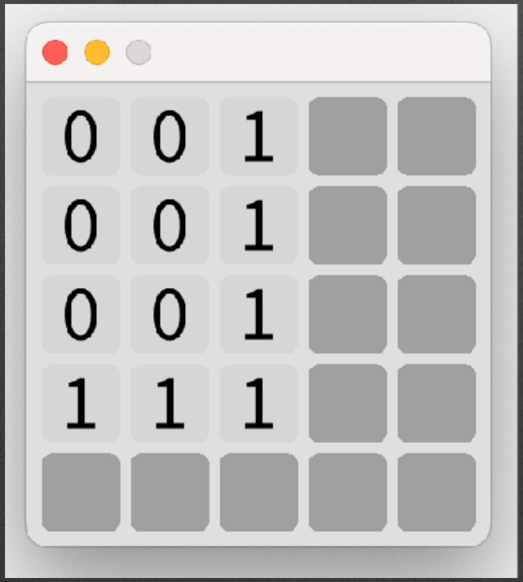
- Gtk3 -> GTK4
- tables/list/trees/drop lists can be improved: right now you have a limited amount of column types to use on them.
- First Toplo version
- Finish with tool migration e.g. Finder
- Calypso migration

New graphics for real

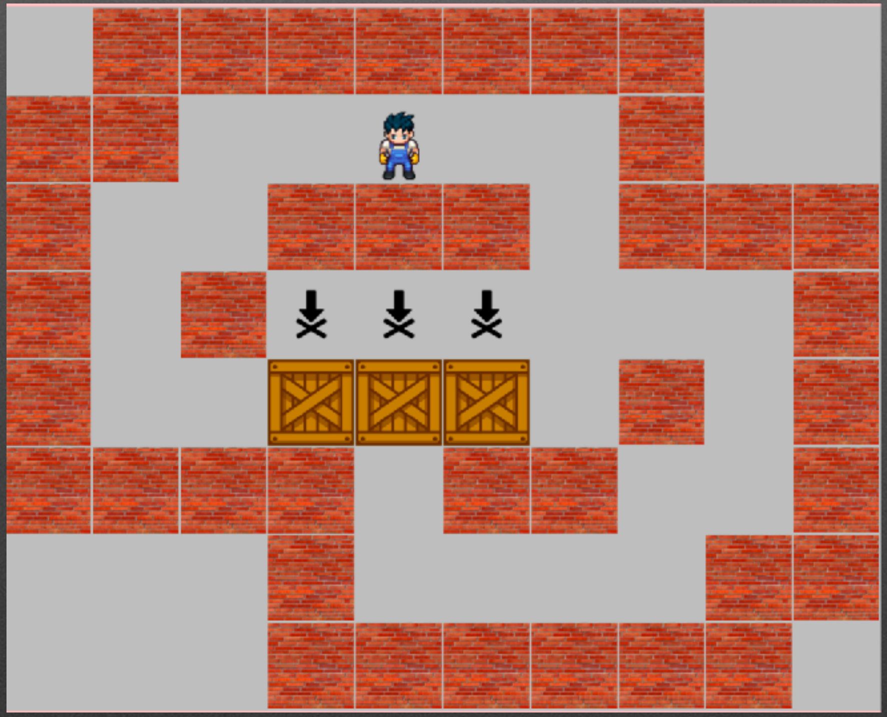
- SDL 20 for events (no more hidden in the VM)
- Failed to deploy HDPI (solution was only for morphic)
- On the back burner
  - Bloc as a new bottom layer
  - Toplo new widget sets
  - Native windows (nearly done in P12 alpha)

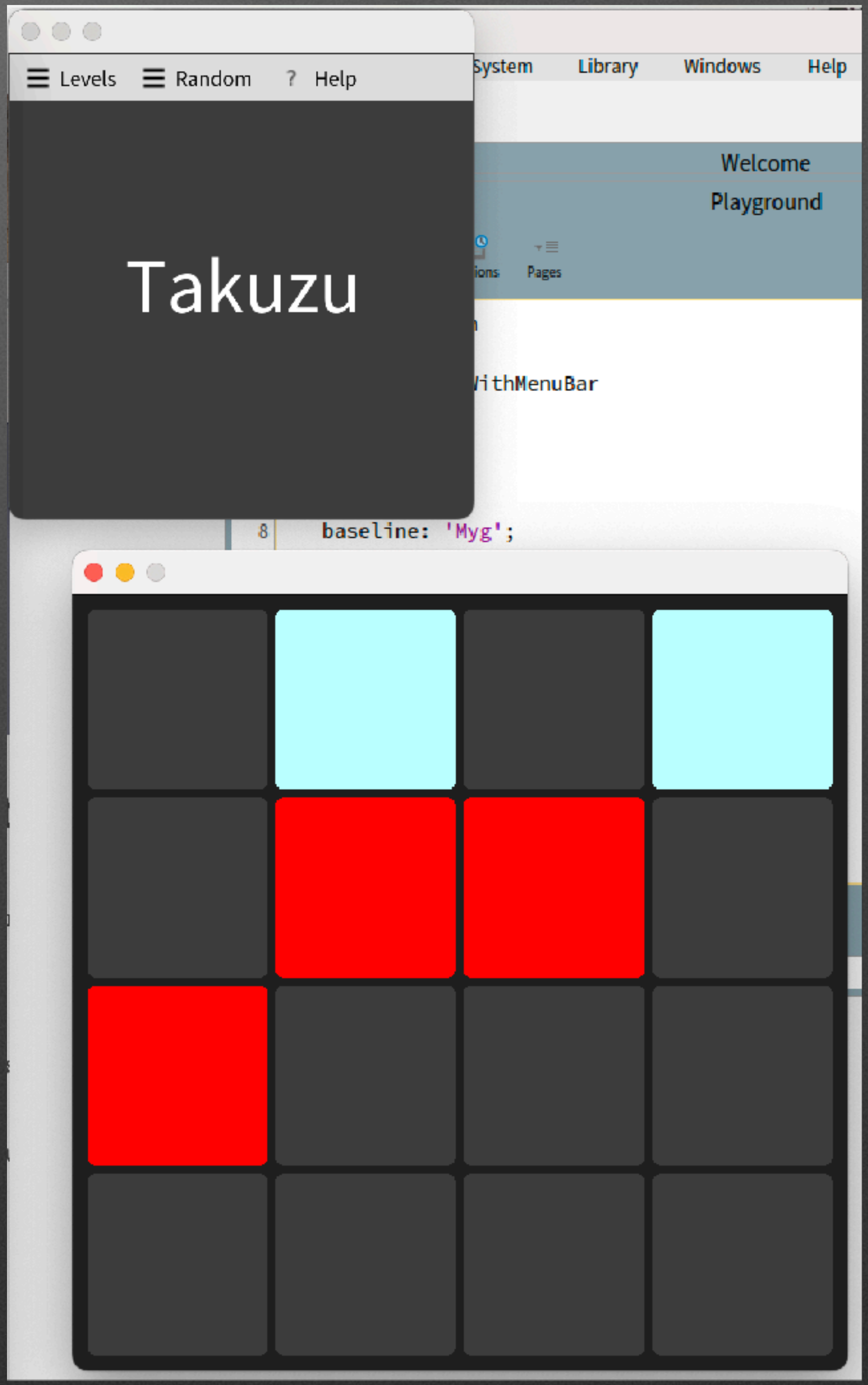
Bloc Update

- Lot of improvements
- See the talk of Martin Dias



0	0	1		
0	0	1		
0	0	1		
1	1	1		



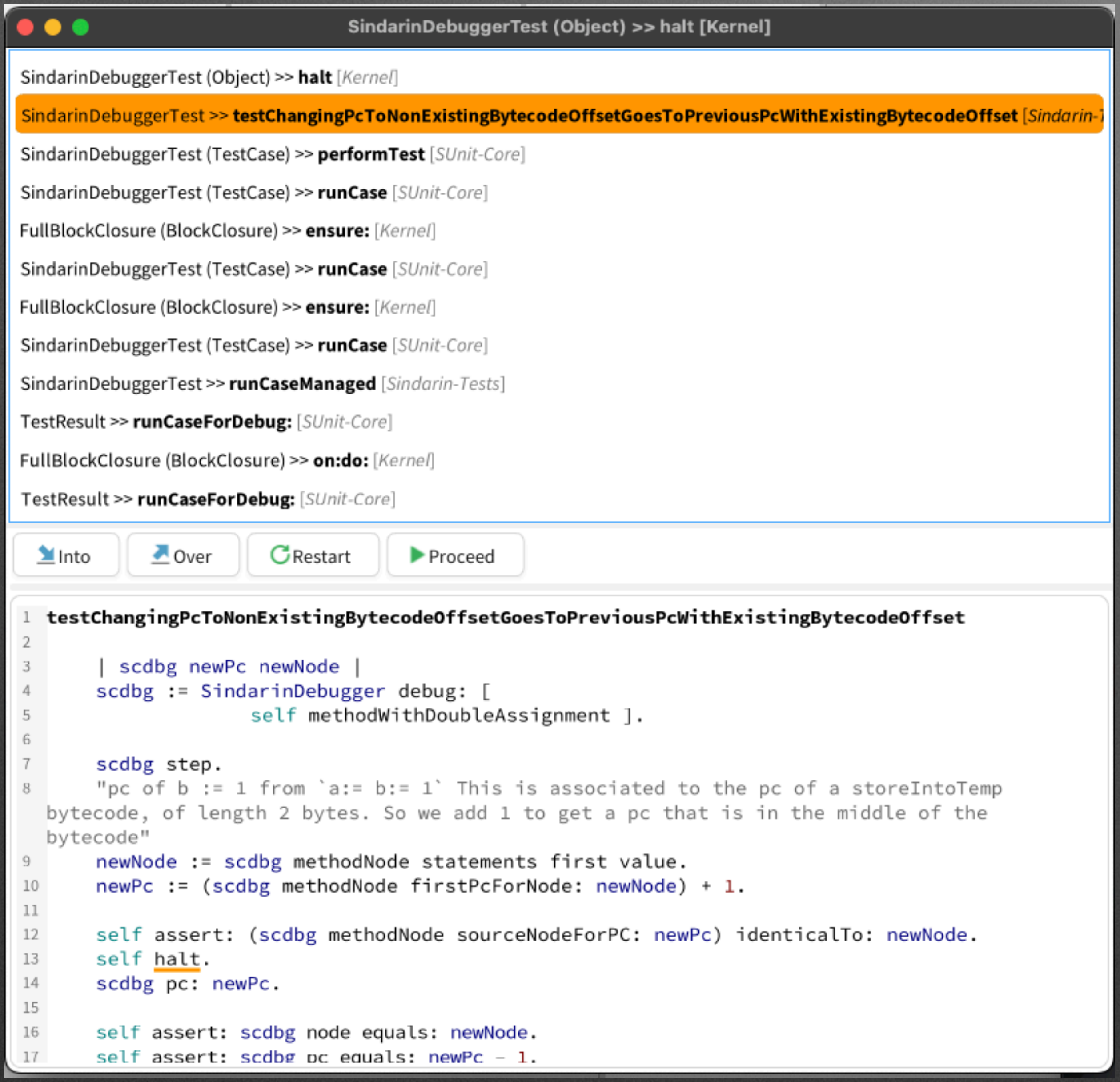


```
baseline: 'Myg';
```

Toplo

- New widget library on TOP of bLOck
- Sponsored by Thales (deployed products in 2023)
- Skins will be based on https://ant.design/
- Started to work on Spec back-end
- Currently
    - All widgets except Tree/Table
    - Skin *first iteration*

Toplo



SindarinDebuggerTest (Object) >> **halt** *[Kernel]*

SindarinDebuggerTest >> **testChangingPcToNonExistingBytecodeOffsetGoesToPreviousPcWithExistingBytecodeOffset** *[Sindarin-*

SindarinDebuggerTest (TestCase) >> **performTest** *[SUnit-Core]*

SindarinDebuggerTest (TestCase) >> **runCase** *[SUnit-Core]*

FullBlockClosure (BlockClosure) >> **ensure:** *[Kernel]*

SindarinDebuggerTest (TestCase) >> **runCase** *[SUnit-Core]*

FullBlockClosure (BlockClosure) >> **ensure:** *[Kernel]*

SindarinDebuggerTest (TestCase) >> **runCase** *[SUnit-Core]*

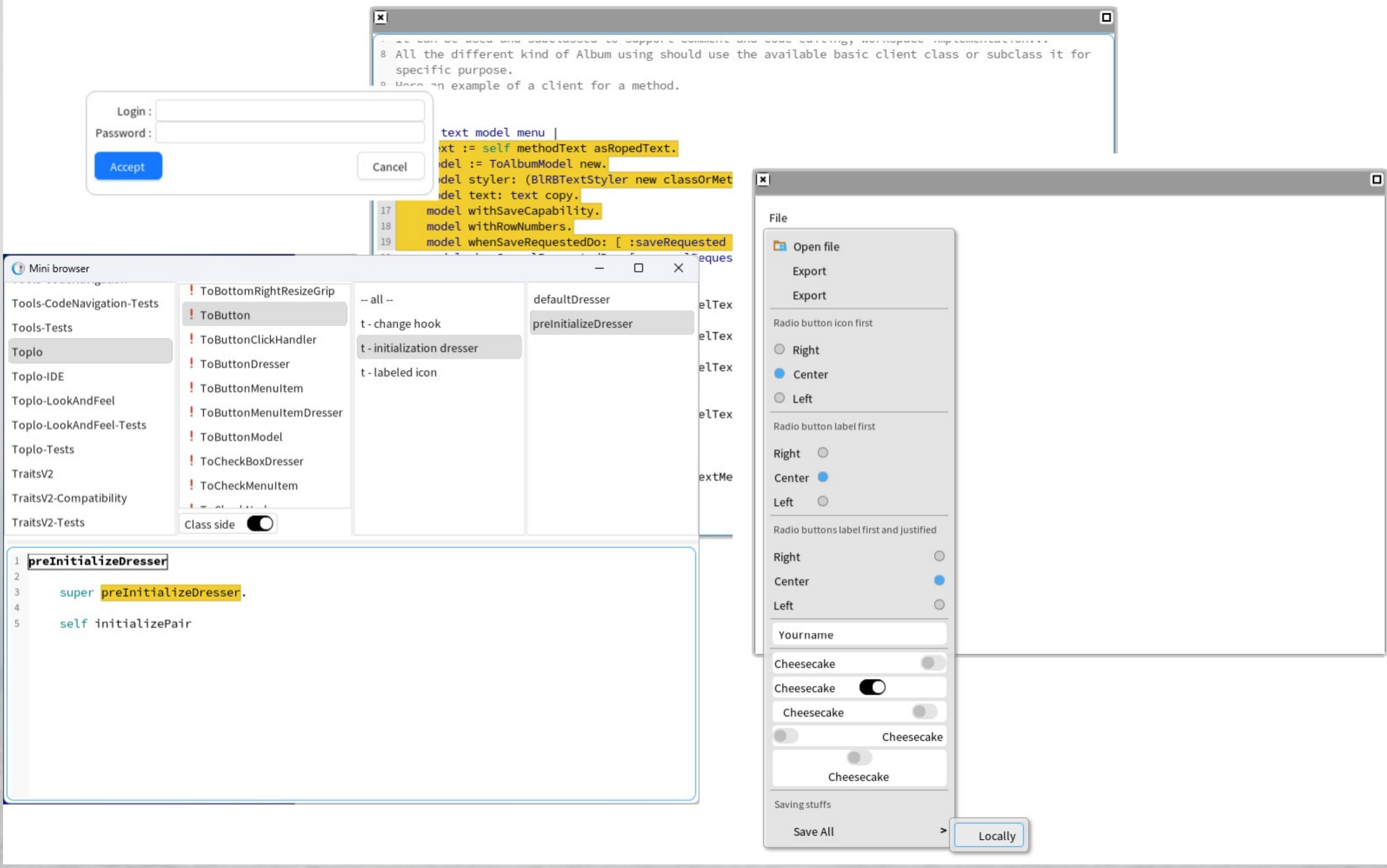
SindarinDebuggerTest >> **runCaseManaged** *[Sindarin-Tests]*

TestResult >> **runCaseForDebug:** *[SUnit-Core]*

FullBlockClosure (BlockClosure) >> **on:do:** *[Kernel]*

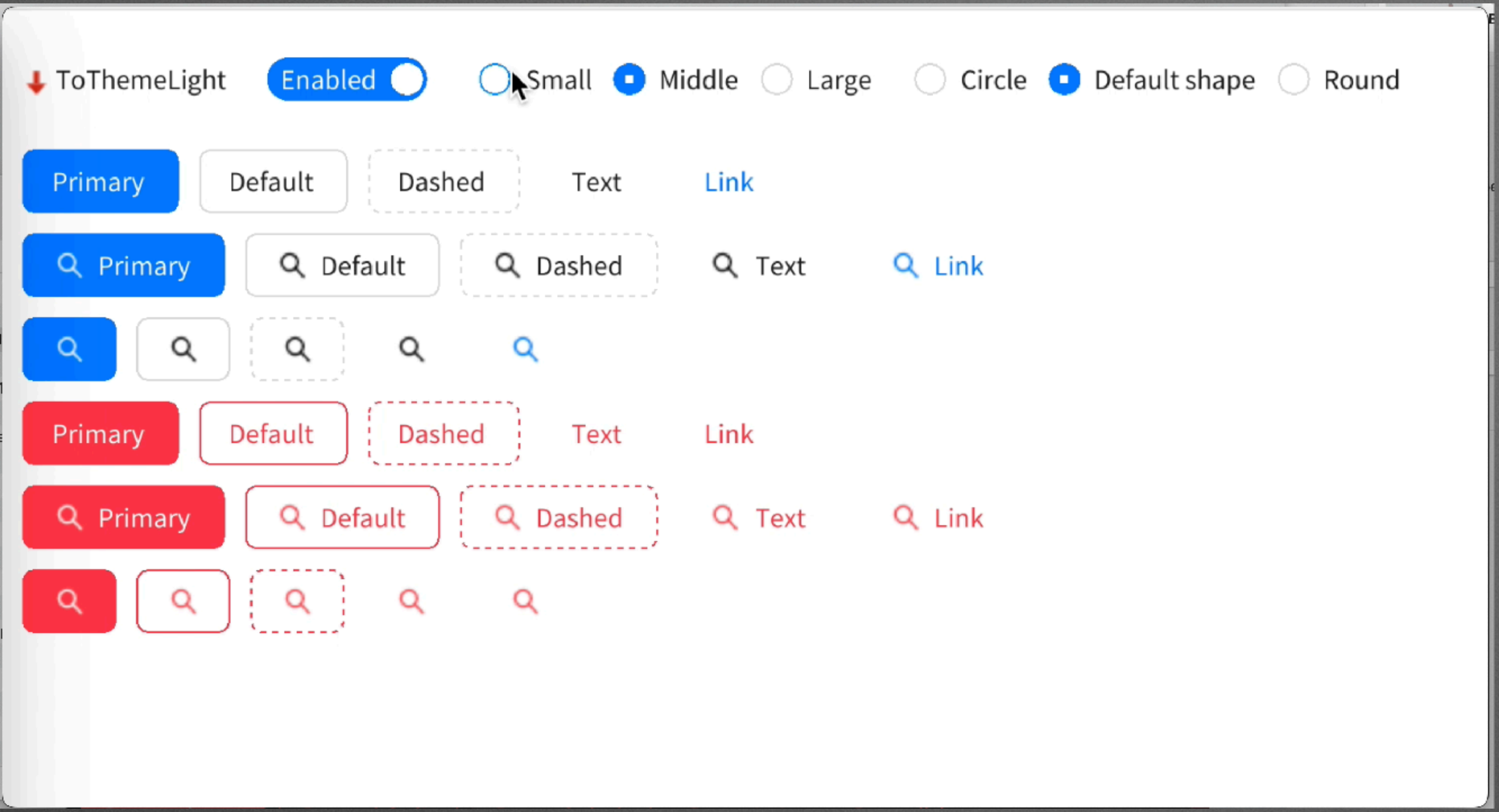
TestResult >> **runCaseForDebug:** *[SUnit-Core]*

```
testChangingPcToNonExistingBytecodeOffsetGoesToPreviousPcWithExistingBytecodeOffset

    | scdbg newPc newNode |
    scdbg := SindarinDebugger debug: [
                self methodWithDoubleAssignment ].

    scdbg step.
    "pc of b := 1 from `a:= b:= 1` This is associated to the pc of a storeIntoTemp bytecode, of length 2 bytes. So we add 1 to get a pc that is in the middle of the bytecode"
    newNode := scdbg methodNode statements first value.
    newPc := (scdbg methodNode firstPcForNode: newNode) + 1.

    self assert: (scdbg methodNode sourceNodeForPC: newPc) identicalTo: newNode.
    self halt.
    scdbg pc: newPc.

    self assert: scdbg node equals: newNode.
    self assert: scdbg pc equals: newPc - 1.
```

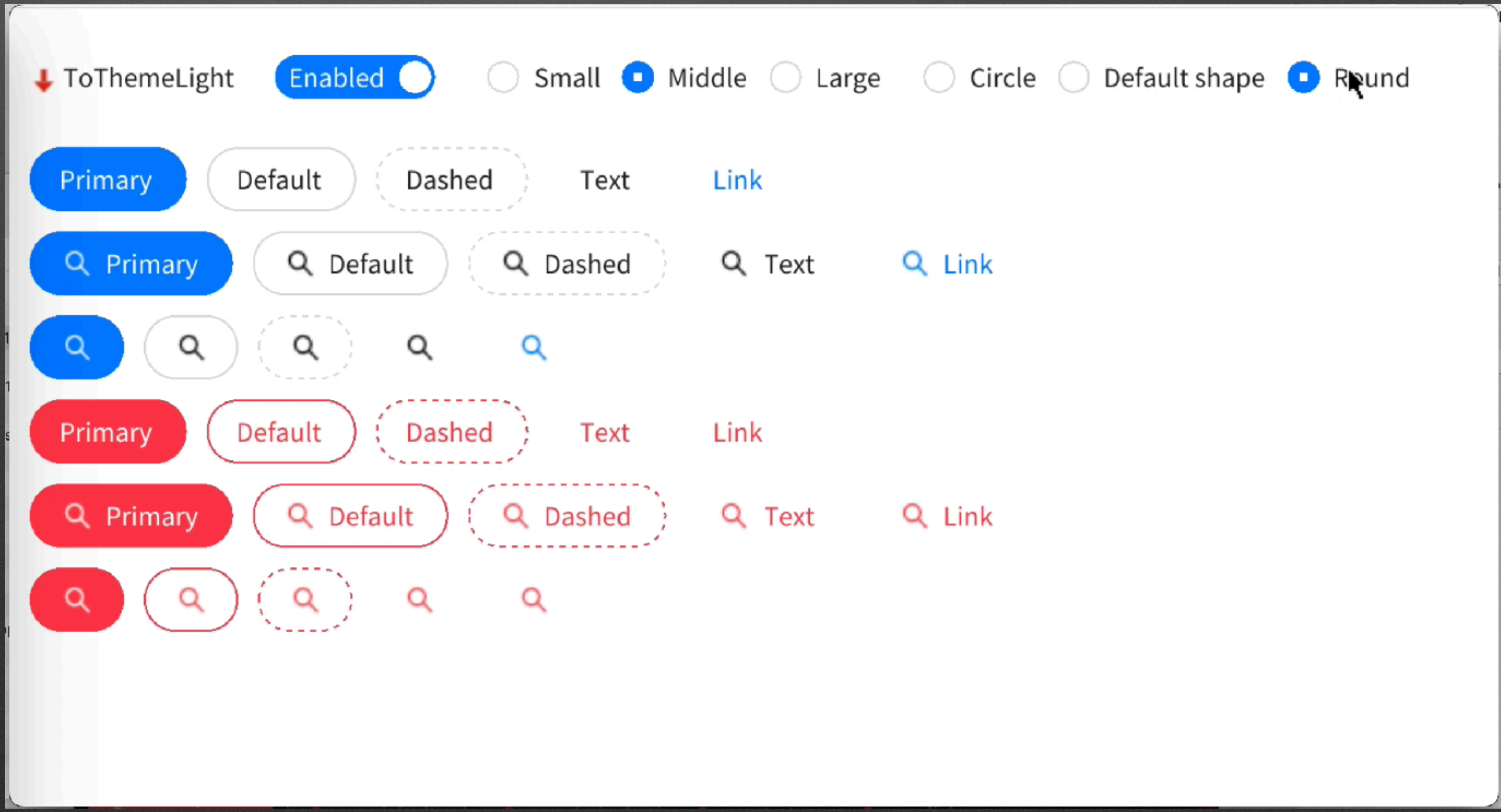


```
All the different kind of Album using should use the available basic client class or subclass it for
specific purpose.
Here an example of a client for a method.

  text model menu |
xt := self methodText asRopedText.
del := ToAlbumModel new.
del styler: (BlRBTextStyler new classOrMet
del text: text copy.
  model withSaveCapability.
  model withRowNumbers.
  model whenSaveRequestedDo: [ :saveRequested
                                        eques
elTex
elTex
elTex
elTex
extMe
```

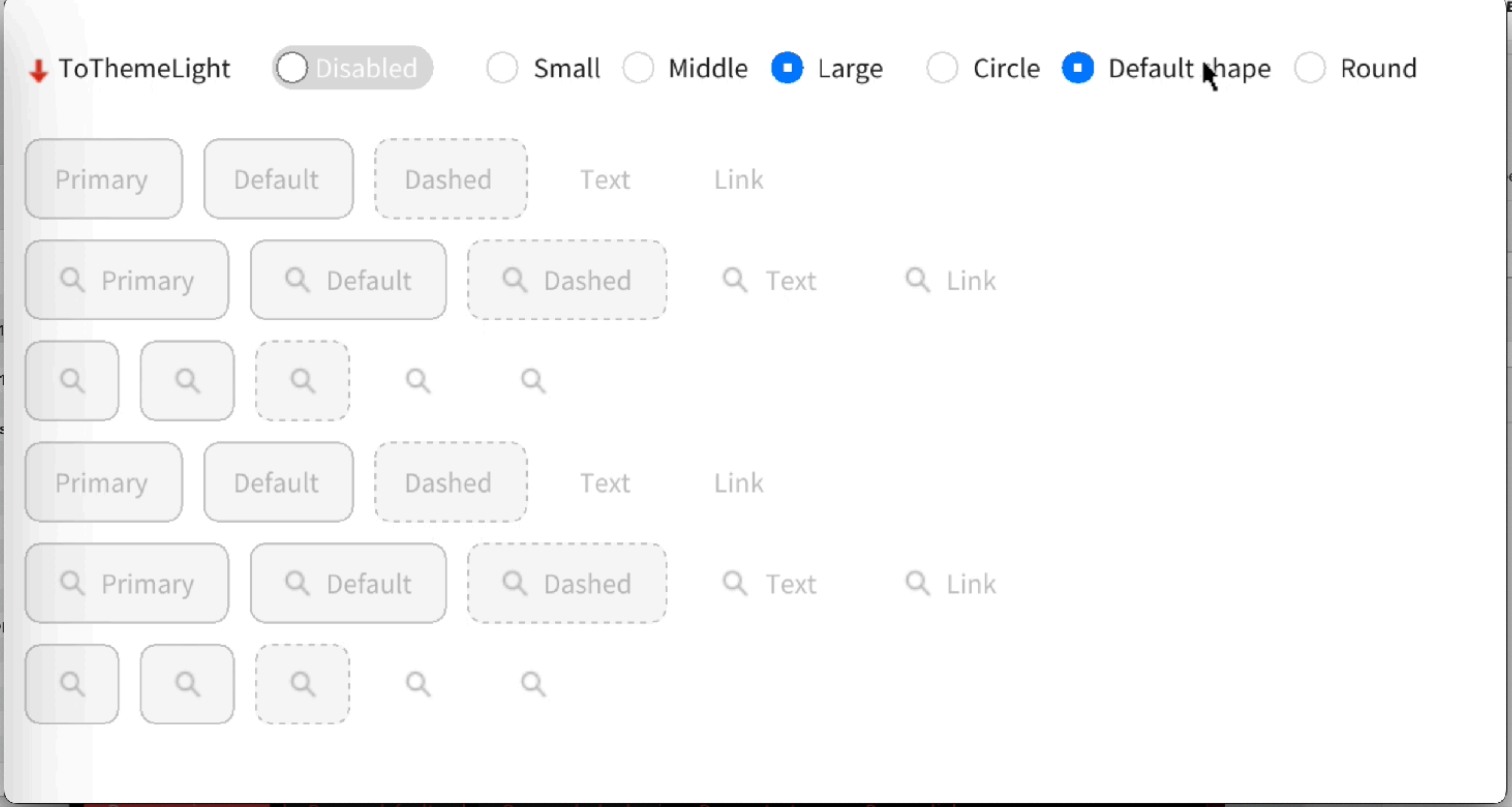
```
preInitializeDresser

    super preInitializeDresser.

    self initializePair
```



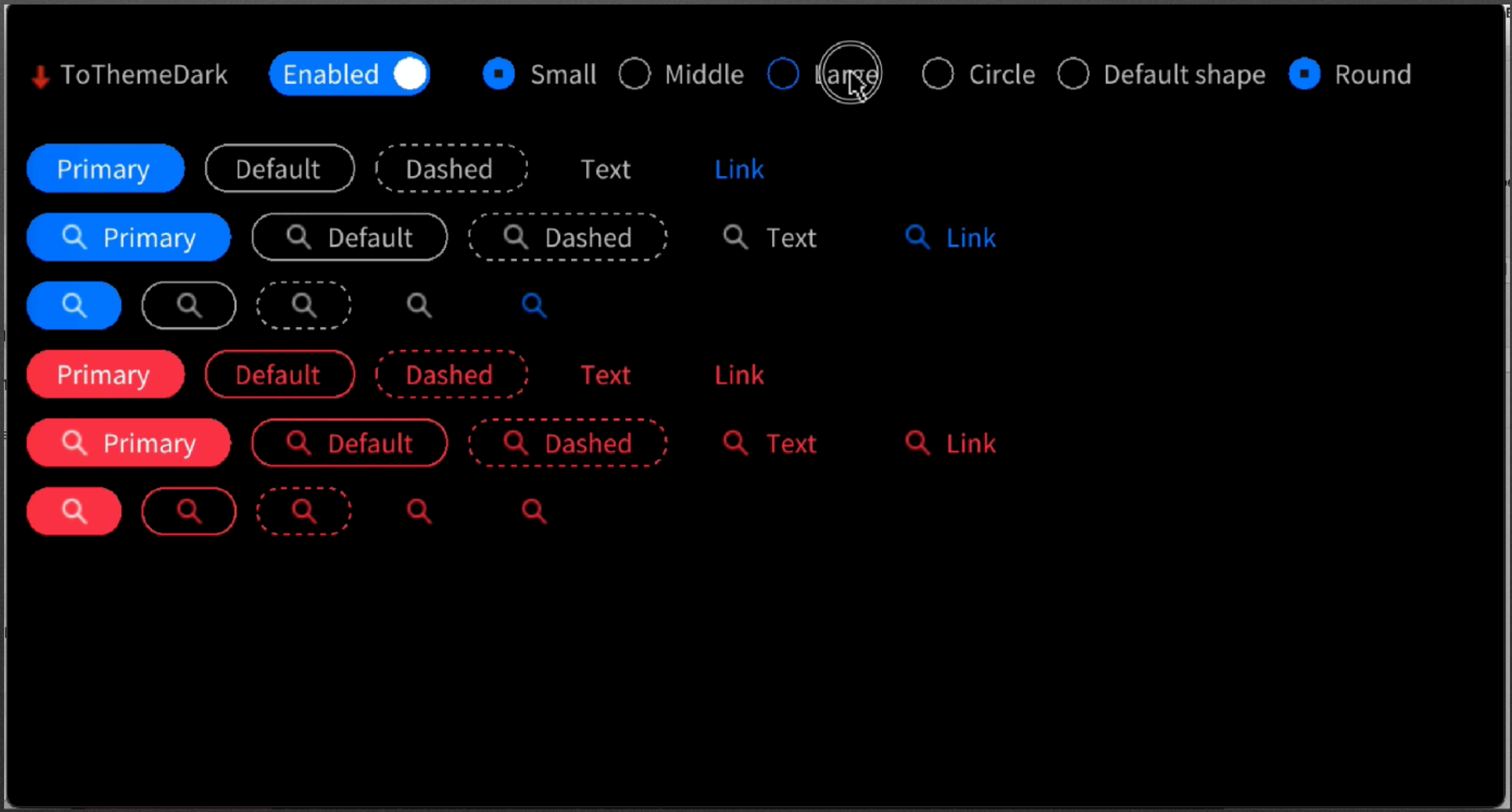
ToThemeLight Enabled ○ Small ● Middle ○ Large ○ Circle ● Default shape ○ Round



ToThemeLight Enabled ○ Small ● Middle ○ Large ○ Circle ○ Default shape ● Round

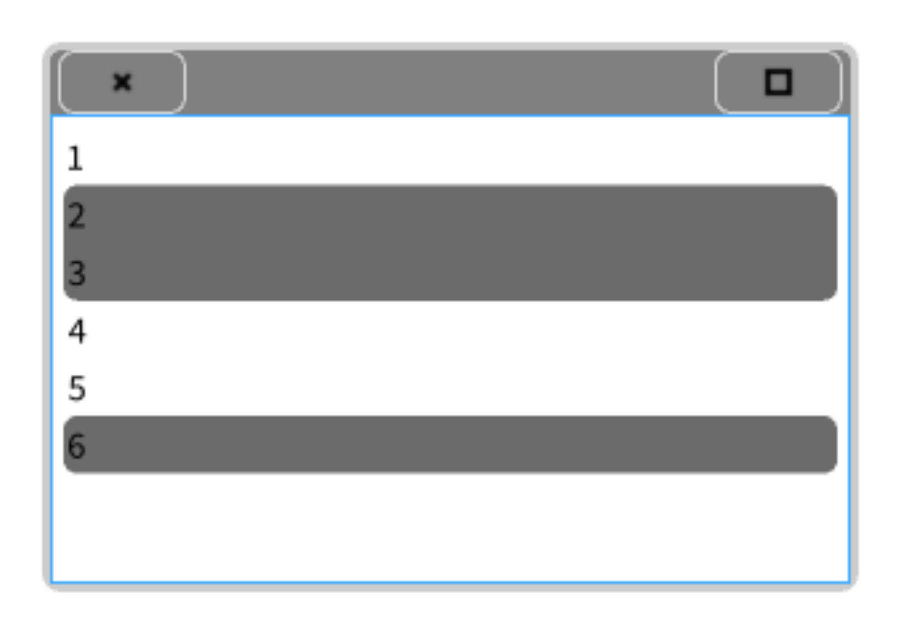


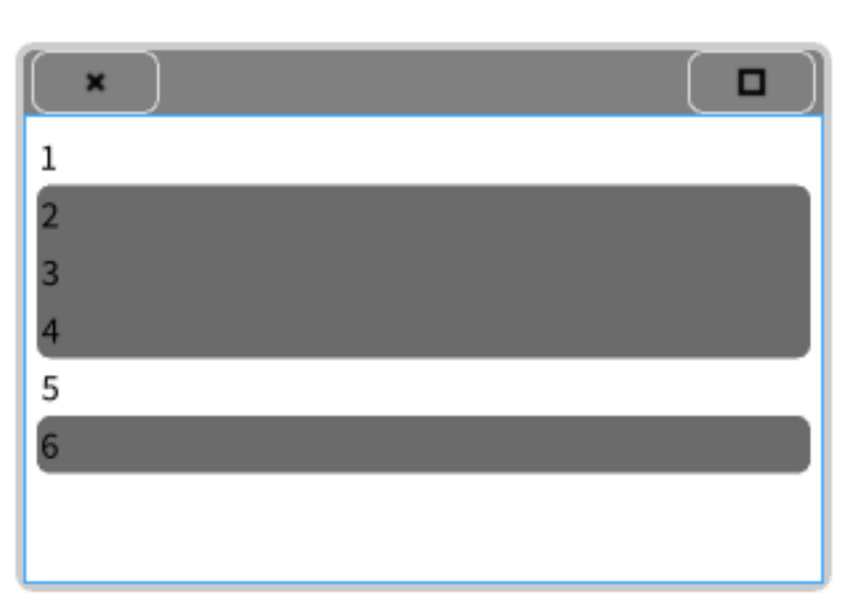
ToThemeLight ○ Disabled ○ Small ○ Middle ● Large ○ Circle ● Default shape ○ Round

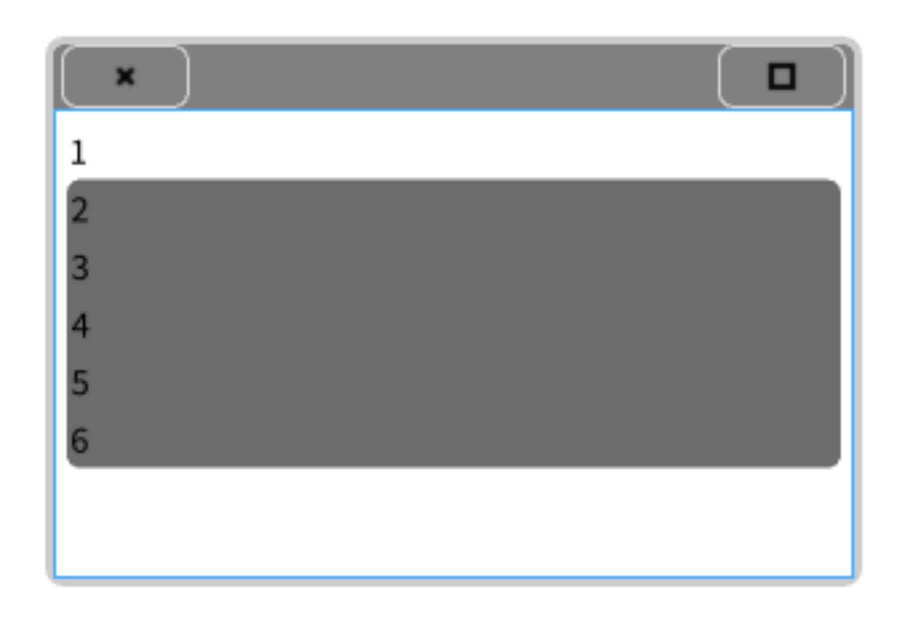


ToThemeDark ● Enabled ● Small ○ Middle ○ Large ○ Circle ○ Default shape ● Round

Real list selection







I want to thank all the contributors

We will continue … :)

A word of teaching

Pharo on Exercism

Thank you guys for the work!
We owe you more than a beer

Excellent Mooc
http://mooc.pharo.org

”I have just completed week seven of the Pharo Mooc …. I have already learned so much ! I have spent the last 20 years or so in software development and, following this Mooc, I realized I hadn’t really grasped the essence of object oriented design”

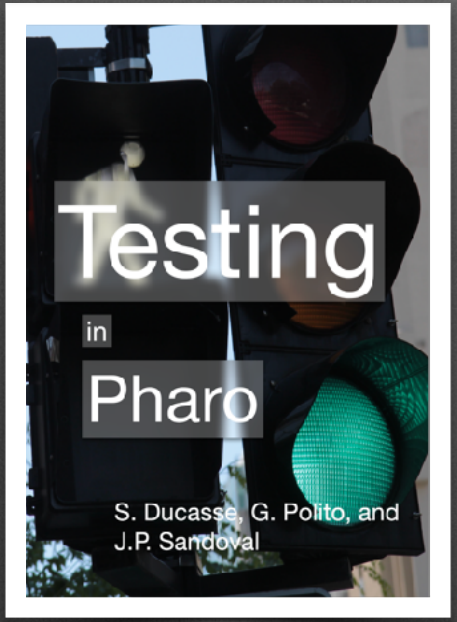
GMJuliet on discord June 2019

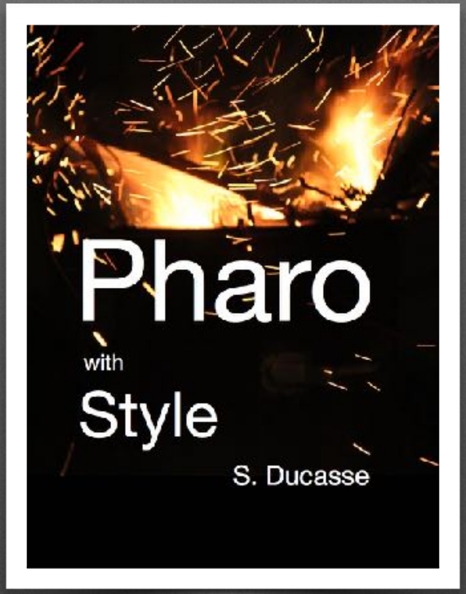
Forthcoming advanced design mooc

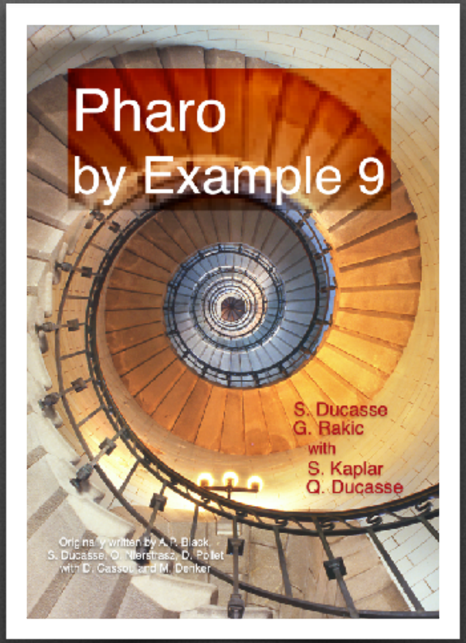
https://advanced-design-mooc.pharo.org/

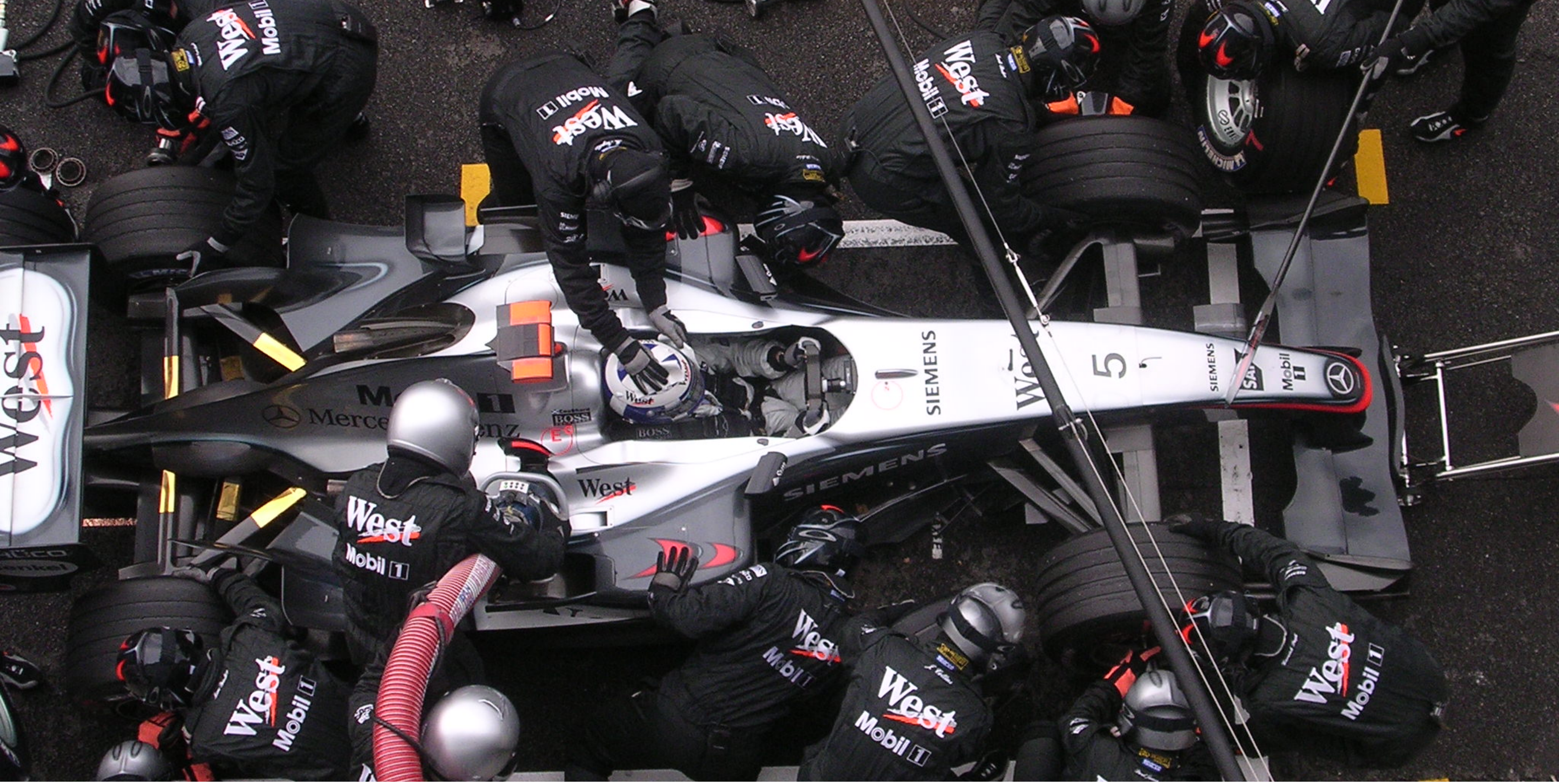
New books

from Keepers of the lighthouse






Pharo
by Example 9
S. Ducasse
G. Rakic
with
S. Kaplar
Q. Ducasse
Originally written by A. P. Black,
S. Ducasse, O. Nierstrasz, D. Pollet
with D. Cassou and M. Denker



Pharo is our vehicle
We improve it everyday

There are plenty of place for improvements

Pharo

is yours

You can get an impact
Pharo is not a closed box

Pharo
ASSOCIATION

LABWARE
Results Count

esug
[ ] EUROPEAN SMALLTALK USER GROUP

Inria

Lam®
RESEARCH

SCHMIDT
Ingenieurbüro für Bauwesen

Lifeware

y
Yesplan
Let's make it happen

OBJECT GUILD

THALES
Building a future we can all trust

ZWEIDENKER

telna

GemTalk™
SYSTEMS

projector
software

inspired!

netstyle.ch

InfOil

FINWORKS
INFORMATION TECHNOLOGY SOLUTIONS

BetaNine
software engineering

TA MÈRE SCRL
BADASS MOBILE DEVELOPMENT

Systems that make sense
Sensus

feenk

cirad

CZECH TECHNICAL UNIVERSITY IN PRAGUE

MINES
Douai
IA
Informatique
et Automatique

FAST
Fundación Argentina de Smalltalk

u b
UNIVERSITÄT
BERN

Toronto
Metropolitan
University

SRA
GROUP

project
uqbar

Evref