Spec 2.0

The next step on desktop UI
Spec 2 is a team work
“How to do a desktop application with Pharo?”

–Most requested feature every ESUG (personal survey)
How can you do an Application today?

- Morphic
- Spec 1.0
- Glamour
- … others
There has to be only one!
(and it has to be easy to use)
Why Spec?

- Modular design
- Testable
- Morphic is low-level: powerful, but too much when doing a regular application (same applies to bloc, by the way)
- Spec can cover Glamour specifics easier than Glamour covering Spec specifics
A Presenter

ComposablePresenter

Layout A

Layout B

Layout C
A composed Presenter
The original plan

- Application
- Spec
- Morphic
The original plan

- Application
- Spec
- Adapter
- Morphic
- Others…
The reality

- Application
- Spec
- Adapter
- Morphic
• Specs mixed with morphs/adapters

• Presenters design are too contaminated with morph design

• Applications need to “jump” layers to do what they need
Spec 2.0!

- The design is already there: Spec 2.0 is about making original Spec design to work properly
- Remove interpreter (because is not needed)
- All Pharo tools needs to be developed with Spec 2.0
  - Having a base UI design for everything
  - Allow to change backends without rewriting the full UI
Some new (or/and revisited) concepts…
Presenters

• A Presenter is the base “presenting” UI mechanism.
  - They can be composed!

• All existing presenters has been revisited (or they will be)

• Some of them has been deprecated! (Tree, TabManager…)

• Some others has even added (Table, TreeTable, ToolBar, …)
Layouts

• Instead one big *do-it-all* layout, small layouts with well defined function
  - Box (vertical, horizontal)
  - Paned (vertical, horizontal)
  - Grid
  - Scrollable
  - Others to come

• Also layouts can be **composed**!

• Current SpecLayout will be deprecated (and removed)
Application

- Entry point of any application (including Pharo)
- Window flow
- Common resources
  - "UI Manager" for standard dialogs
  - Icons/images
  - Theme
- Backend
Backends

- Morphic Backend
- Gtk3 Backend
- Future: Bloc backend
Demo(s) :)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>halo</td>
<td></td>
</tr>
<tr>
<td>copyToPasteBuffer</td>
<td></td>
</tr>
<tr>
<td>savePatchFrom</td>
<td></td>
</tr>
<tr>
<td>balloonHelp</td>
<td></td>
</tr>
<tr>
<td>generateMouseEvent</td>
<td></td>
</tr>
<tr>
<td>obtainHalo</td>
<td>&quot;Used for transferring halos between hands&quot;</td>
</tr>
<tr>
<td>self halo == aHalo</td>
<td></td>
</tr>
<tr>
<td>ifTrue: [ ^ self ].</td>
<td>&quot;Find former owner&quot;</td>
</tr>
<tr>
<td>self halo: aHalo</td>
<td></td>
</tr>
</tbody>
</table>
obtainHalo: aHalo

"Used for transferring 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
Welcome to Star Wars: Legends of the Jedi

(©)ntact 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, Rajan, and Miros
(G)ame Owners: Rajan & Waldo
...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]
時
Spec 2.0 summary

- Easy way to create REAL desktop applications!
- Multiple backends (starting with Morphic and Gtk3, Bloc will be added later)
- For Pharo: Easy migration from one backend to another
- Coming SOON (™)
  - Pharo 8
  - There will be a migration path (most cases are compatible or straightforward)
Extra: Why GTK3?

• Cross platform
• Open source (LGPL)
• Mature
• Popular
• C bindings
Extra: Missing parts

- Finish missing/incomplete presenters
  - Rich text presenter, …
  - Some layouts
- Stabilisation
- Pass on API
- Pass on features
- Headless VM and image-side parts
Extra: The place of bloc

- Replacement of Morphic (not Spec)
- Low level: Applications will still need some higher level layer