Grafoscopio User Manual

Offray Vladimir Luna Cárdenas

24 September 2017

Contents

1   Important information to start with

1.1   A small presentation

This manual documents the basics of Grafoscopio, a tool I'm cocreating[1] to make reproducible research and literate computing, which allow authors to intertwine prose, code, data and visualizations into storytelling, and readers and coauthors can verify, collaborate on and extend the document claims and artifacts. The context for that cocreation is my PhD in Design and Creation in the University of Caldas (Manizales, Colombia) and in our local hackerspace HackBo (Bogotá, Colombia), but we are making something that moves beyond and between frontiers and our hope is to potenciate that.

This document started as a draft for a simple `README` file, following the recomendations given in the Jornal of Open Source Software (JOSS), but

---

[1]See the acknowledgements to look for other people that make Grafoscopio possible. Hopefuly this form of cocreation is just the beginning and will become more plural and potent in the future.

I entered in some kind of "writing frenzy", and when I stoped, I had 40+ pages of documentation that became this manual, which is a way to introduce Grafoscopio to a wider English speaking/reading audience.

I think there is a lot happening beyond the usual places and languages, here in the *Global South*: in garages, hackerspaces, makerspaces, favelas, barrios, streets, in Portuguese, Spanish, Hindi and away of indexed journals and their narratives. Away of universities, research centers, software fabrics and "publication scores". JOSS is an interesting enactive hack/critic to hegemonic ways of seeing knowledge and its validation, that visibilizes other creations and artifacts beyond the paper, by given them a paper facade and we need more hacks like this, to make visible non-hegemonic knowledge practices, places, artifacts, cultures and languages.

Grafoscopio is an artifact to bridge, explore and communicate ideas and practices to and from a broader and more diverse group of people and to amplify their voices with digital technology. I hope you find the tools and ideas presented here interesting and they become a bridge to alternatives that resonate with you.

1.2   Acknowledgements

As Nadia Egbal (2016), Benjamin Mako Hill (2013) and others have pointed, the majority of the Free Libre Open Source Software is created and mantained mainly by one or two individuals, and in that sense Grafoscopio, follows the rule and not the exception, with me as the main author of Grafoscopio and its companion packages (`Dataviz`, `Pubiblio`, `Fossil` and `Grafoscopio-Utils`) Despite of that, I think that Grafoscopio has been cocreated in a pretty particular sense: our hackathons and workshops have given me a strong sense of priority and flow, and that requires a lot of commitment and openess from most of the participants. Also, I have received support from family, friends and tech communities in several ways: long coffe talks, code snippets, proactive workshop documentation and note taking, jokes, critic & challenges, travel help, proof reading blog entries, and even small and flexible loans (and is really a bless having them coming from family and not from banks). In that sense each creation requires a lot of people to see the light and I have been lucky enough to count with all of them.

Here is a (probably incomplete)[2] list of people that helped me to make Grafoscopio possible:

Philippe Back, Claudia Baez, Carlos Barreneche, Alexandre Bergel, Luis Alejandro Bernal, Andrés Calderón, Hilda Cárdenas, Fernando Castro Toro, Sebastian Castro Toro, Ben Coman, Jose David Cuartas, Stephan Eggermont, Johan Fabry, Yanneth Gil, Tudor Girba, Nicolai Hess, Diddier H., Camilo Hurtado, Luis Linares, Divian Luna, Miltón Mamani, Rafael Medina, Iván Pulido, David Salvador, Serge Stinckwich, and Peter Uhnák.

Also, several places and communities were an active part in this project: HackBo hackerspace, mutabiT, Pharo, Moose and Agile Visualization communities, HiTeC Lab of the Los Libertadores University and Object Profile.

To all of them, my sincere thanks.

1.3   Reading conventions

This document follows the following conventions:

- *Italics* is used for emphasis and for Grafoscopio/Pharo special terms. For example: "Write this in the *playground*".
- **Bold** is for strong emphasis notes and reminders, like in "**Important note**".
- `Verbatim` is for representing code snippets and commands, keys and file names. For example: "Open the file named `manual.ston`".
- Code lines are in verbatim and numered, which makes easier to point to specific parts and to know when a code block splits across sucessive pages.
- `Ctrl` and `Cmd` represent the "Control" and "Command" keys, respectively. The former is avaible on Windows and Gnu/Linux platforms and the last one is available on Mac platform. When we use that key in combination with other that needs to be pressed at the same time, we use + and the other key, so `Ctrl + s` represent the simultaneos pressing of the keys `Ctrl` and `s`. Keyboard shorcuts that contain `Ctrl` on Windows and Gnu/Linux, can be replaced by `Cmd` on Mac and to ease

---

[2]If I made and error on omission in your name, please use the community contact forms explained at "Community Guidelines" section to let me know.

reading we don't repeat each shorcut on each platform. Sometimes we will put the keyboard shortcut in parenthesis, just after the action it referst to, like in "Open the spotter (`Shift + Enter`)".

- When we refer to menus and submenus in the Graphical User Interface (GUI), we will use `>` as an indicator of the menus hirarchy. For example, writing `Update > Grafoscopio`, means to choose the `Update` menu and then the `Grafoscopio` submenu in it.
- The PDF and HTML versions of this document include clickable links and words, that are shown in blue, for example the following words are a clickable link to the Grafoscopio web page.
- Bibiographic references use the usual convention of being cited between parenthesis, like in: "(author year)".

1.4 Document and software versions

By convention, this manual has the same version of the software it documents. So the version number you see on bottom left side of each page, corresponds also to the Grafoscopio version. Sometimes the manual is extended or corrected, while referring to the same version of the software, so an alphanumeric smaller identifier is added next, in square brackets. Version plus revision identify without ambiguity this manual[3], which is useful for changing digital artifacts to report bugs, fixes or just to know at which point of their history you are located.

Because we follow a rolling release model for Grafoscopio and its related packages and projects it is possible that the software and the documentation get out of sync. Update them to their latest versions using the `Update` menu from the *docking bar*, as will be explained later.

Grafoscopio is packaged with other companion software, that is installed automatically with it. This is the info about this software bundle (taken from the present `ConfigurationOfGrafoscopio` source code):

```
1  version14: spec
2      <version: '1.4' imports: #('1.2-baseline' )>
```

---

[3]For the curious, that code corresponds to the cryptographic hash of the source markdown file used to produce the PDF or HTML version of this document (you will learn more about them in the "Exporting" section of this manual). You can use this hash to locate the exact version of such file in its timeline.

```
3
4      spec for: #'common' do: [
5          spec
6              blessing: #'stable';
7              description: 'version 1.4';
8              author: 'Offray Luna'.
9          spec
10             project: 'Ston' with: #stable;
11             project: 'Fossil' with: #stable;
12             project: 'Roassal2' with: #stable;
13             project: 'WebBrowser' with: #stable;
14             package: #'Grafoscopio-Utils' with: 'Grafoscopio-Utils-OffrayLuna.19'
15             project: 'Dataviz' with: '2.2.6'.
16         spec
17             package: #Grafoscopio with: 'Grafoscopio-OffrayLuna.298';
18             package: 'Spec-Glamour' with: 'Spec-Glamour-johanfabry.2';
19             package: 'Pubiblio' with: 'Pubiblio-OffrayLuna.2'.]
```

1.5   On commons, copyright and copyfarleft

Grafoscopio is covered with the same MIT license as Pharo. This document is covered by the P2P license. Both are pretty liberal licenses that grant you a plethora of rights to use, modify and make profit of Grafoscopio and its documentation, under certains conditions, but I think that documentation and software do not need to have the same license, or offer the same rights even in the case of Grafoscopio, where *interactive documentation is closer to being a form of software.*

I think that a deeper discusion on licences and the protection and expansion of knowledge as a commons is needed and this licenses difference reflect that. You can see a full copy of both licenses, MIT and P2P, included in the "Licenses" section of this document.

2   Grafoscopio for what and for whom?

2.1   This tool and you

Grafoscopio is a moldable tool for literate computing and reproducible research, developed on Pharo live coding and computing integrated environment, in the context of a PhD research in a hacker space of the Global South (HackBo in Bogotá, Colombia), that is being actively used, developed and documented. We will expand on the points of the previous definition.

- Being moldable (Girba, Chis, and Niertrasz 2014) means that is easy to adapt the tool to several problems, which follows the opposite popular path of trying to force several problems into a predefined tool. Tools change us. So we need to change them back to express our concerns and to help us in dealing with complex issues and problems in a flexible way.
- Literate computing (Perez and Granger 2015) is a way of intertwining prose, code, data and visualizations to make data based storytelling, experimentation, exploration and documentation in diverse broad fields like academia, journalism, research and (h)ac(k)tivism. Grafoscopio allows the creation of literate computing structured interactive notebooks, that take the form of a tree-like programmable document.
- Research claims and findings and supporting data and visualizations can be integrated in interactive notebooks with historic traceability, allowing reproducible research.
- Because of the continuity and uniformity of the Pharo environment (Black et al. 2016), Grafoscopio blurs the distinction between code, data, document, application and IDE[4] and tries to blur also the distinction between interactive documents authors and software developers.
- From the particular context where is being developed (HackBo hackerspace and a PhD research on Design and Creation), Grafoscopio is concived as a empowering simple and self contained *pocket infrastructure* (that work off-line/on-line from USB thumb drives and/or low resources machines (Luna Cárdenas 2014)), wich states a critical approach to exclusionary ideas about data, coding, research, and their practicioners, places, and supporting tools and infrastructures. In the

---

[4]IDE: Integrated software Development Environment

Grafoscopio web page, we showcase several projects aligned with such critical approach and the idea that technology is mostly not neutral, wich is also reflected in some parts of this text and in the fact that we're opinionated about technology and its uses. I hope you find our technology and themes choices empowering and reavealing of alternatives.

Grafoscopio and its main companion package, Dataviz, are intended to be used by learners and researchers in several fields: academia, journalism, activism, hacktivism and for anyone interested in open reproducible research and data storytelling backed by data and agile visualizations (Bergel 2016). This document assumes that you are such person. We will introduce the general features of Grafoscopio and point to several external and internal resources to complete your panoramic view of the ecosystem that let you deep your knowledge.

We included introductory resources to learn Pharo and data visualization, processing and storage related technologies (see the `Help` menu), and the Grafoscopio web page (see figure 1) shows several examples of how to use them for specific projects: Panama Papers as reproducible research; open community innovation in access to medicine information; Twitter data selfies; specific domain visualizations for medicine information; open, garage and citizen science and research. *This whole manual was also created using Grafoscopio* and is also an example of the type of documents you can create with this tool. We hope to inspire you to create and publish your own projects.

This document, by not trying to be comprenhensive, is a first invitation to know the Grafoscopio environment and to deep your knowledge with the use of it and other related resources. You will see that, despite of being a manual, it includes pretty practical examples and invitations. That is because I think that learning something new is more like reading a map that reading a manual: you make first a panoramic view of where you are and where you want to be, and take a practical approach to making your tasks and reaching your destination.

No prior knowledge of programming is supposed to follow this manual.

Important note > *A prototype pointing to future possibilites* | Despite of being pretty usable, you will see that Grafoscopio is not totally finished, and this shows in a few spots of the Graphical User Interface (GUI) that "point
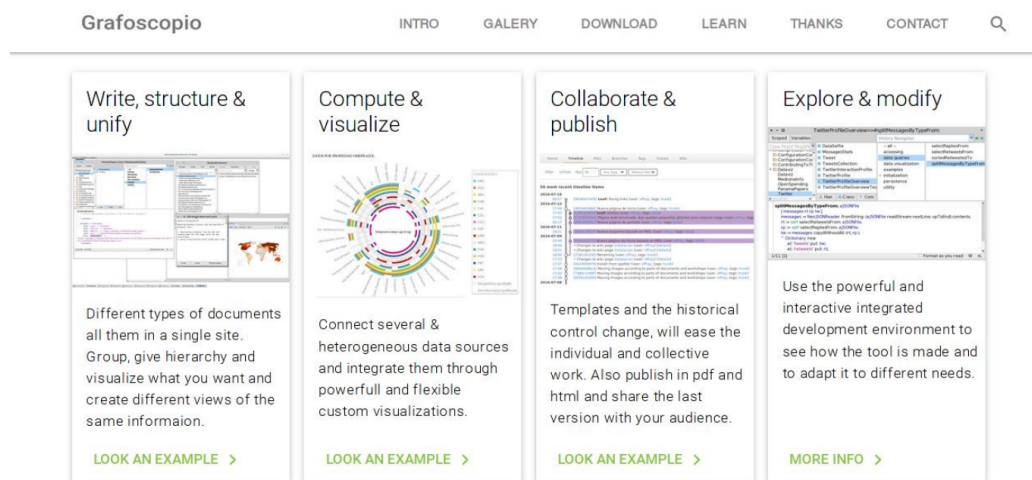
Figure 1: Detail for the Grafoscopio English web page.

to the future", towards funtionality still to be implemented. It's an unusual approach, but I think that is important to convey some sense of possibility, and work to be done in the GUI, instead of a fully polished "product" or a GUI that hides what is not ready. This conviction comes from the hackathons and workshops where we worked and evolved Grafoscopio, while the workshop was happening(!), thanks to the dynamic, moldable and continuous nature of the Pharo live coding environment. Blurring the distinction between interactive documents authors and software developers, means to put the whole environment at their dispossal, and to show the community that they can be part of this future possibilities, and that is why we take this unusual approach to GUI.

Where the GUI is more a remainder for the future, I will point that using the **TBD** remark (for To Be Done).

2.2   Place in the ecosystem

2.2.1   Similar tools

Grafoscopio is similar to other tools and has been inspired by many of them, while is trying to bring also new possibilities, by combining different ideas, di-

verging from others, puting "parallel" ideas into dialog and, hopefully, bringing new ones. Here we talk about the similarities and differences with other tools.

- Like Jupyter, or Zeppling, Beaker or nteract, Grafoscopio provides interactive notebook functionality, but it is focused only on Pharo code right now, instead of being a "language neutral" notebook, (this could be a feature for the future) and is able to use Pharo bindings to integrate and communicate with other data languages, like R, SQL or J. Grafoscopio is a multiplatform (Mac, Windows, Gnu/Linux) desktop application (like nteract, or Electron Beaker), instead of a web one (like Jupyter, Zepelling or Beaker), providing a simple, extensible, powerful, self-contained and portable environment for interactive computing, (as said it can run from a USB thumb drive, modest computers and anything in between and beyond).
- Grafoscopio organizes documents in a tree like metaphor, also called the *outline*, or the notebook, that is interactive and programmable, like Org Mode, Leo Editor, TeXmacs or Pollen and share with them the idea that the "document is the program"[5] (or a programable container of small chunks of programs and scripts). Also, the document author, can define custom tags that can be used to traverse the document tree and produce multiple customized views and outputs from a single document. A single notebook can contain short or book size interactive documents (this full manual is in a single Grafoscopio notebook).
- Like Jupyter Lab, Grafoscopio environment supports needs beyond the notebook. Grafoscopio achieves this by leveraging the advantange of the extensible Pharo computing environment and ecosystem, where it inhabits, with powerful tools for several computing taks, beyond and complementary to interactive documentation and reproducible research: GUI bulding, data processing and visualization, unit testing, code repositories and source management, among others. It could be said that Jupyter Lab and Grafoscopio followed opposite paths, the first started in the Jupyter notebook, and is trying to become and IDE, and the second started in the Pharo live coding IDE and it bringing

---

[5]The idea of the "document is a program" is a paraphrasis of "the book is a program", stated in the Pollen documentation, which is a short phrase to express a powerful idea about blurring the disctinction between the document and the program, that is present in several programs, like TeXmacs, Leo, Org Mode, and, of course, Grafoscopio.

> interactive notebooks functionality. In some sense, one is already in
> the future of what the other can be.

- Grafoscopio uses the Roassal agile visualization engine, to build inter-
  active visualizations and export them to the web. Roassal provides
  similar functionality to other visualization engines and toolkits like
  D3.js, RaphaelJS, Processing or Flare, but, by being part of the Pharo
  live coding environment, it invites to a more explorative and dynamic
  building of visualizations in an agile way.
- At the moment, notebook sharing, collaboration and publishing in print
  (PDF) and web (HTML) formats is supported, but in the future we
  hope to provide advanced interactive notebook publishing features in a
  distributed p2p fashion (see next section for the techologies that enable
  this).

Important note > *On "single language" notebooks* | There are other "sin-
gle language" notebook like The Gamma, that bridge to the world, using a
common language like Javascript. Grafoscopio bridges to the world via For-
eign Function Interface (FFI) that allows it to be extended by pretty much
anything out there and is the way the R, SQL or J extensions are allowed.
You will find, though that Pharo is a pretty expressive language and envi-
ronment and that it shines at the agile and custom visualization techniques,
thanks to its consistency and liveness and to the Roassal engine. My invita-
tion is to make Grafoscopio your home for your data explorations and extend
it as needed with external tools and languages, as we have shown in examples
like the Panama Papers one. You will see that it provides a unique place for
interactive computation and data storytelling and visualization and adds to
your workflow as a place where most on your work can happen and as a hub
for external tools.

2.2.2   Technologies behind

Grafoscopio tries to become a simple, understandable, moldable, versatile
and flexible tool thanks to the power of Pharo environment and ecosystem
and the combination with mature external and internal frameworks and tools.
It uses:

- Internal tools and frameworks:

- GT Tools and Spec for embeddable code playgrounds, GUI and interactive notebook nodes.
- Roassal for data visualization.
- STON for a light data storage and a human friendly notebooks format.
- NeoJSON for interacting with structured hierarchical JSON data.
- Citezen: for reading and exporting bibliographies to the BibTeX format.
- Fuel: For medium data storage and objects serialization.
- UDBC: For connection and management of external data bases.

- External tools and frameworks:

  - Fossil SCM for collaboration, publication and traceability of the documents history (including this very manual).
  - Pandoc for exporting to printing (PDF) and web (HTML) formats.
  - SQLite for storage and management of tabular data, for the `Dataviz` companion package.

Despite of trying to provide a friendly, cohesive and empowering user experience (UX) by integrating default external minimalist and/or self-contained tools into the data exploration and document publishing workflow, other external tools could be integrated (Git, more data bases, including NoSQL, other exporters and light markup languages and so on).

3 Installation instructions

If you want to install Grafoscopio on Pharo 6 or 6.1, there are two ways of doing: via the Pharo catalog or via running a script from the playground. Both suppose that you have already installed and running Pharo for your platform (Windows, Gnu/Linux or Mac) from its download page, and will be explained below, from the easiest one to the far easy ones. Also both of them use the Monticello package manager, so dependencies are managed automatically for you, making the procedures really short, even for the script based one.
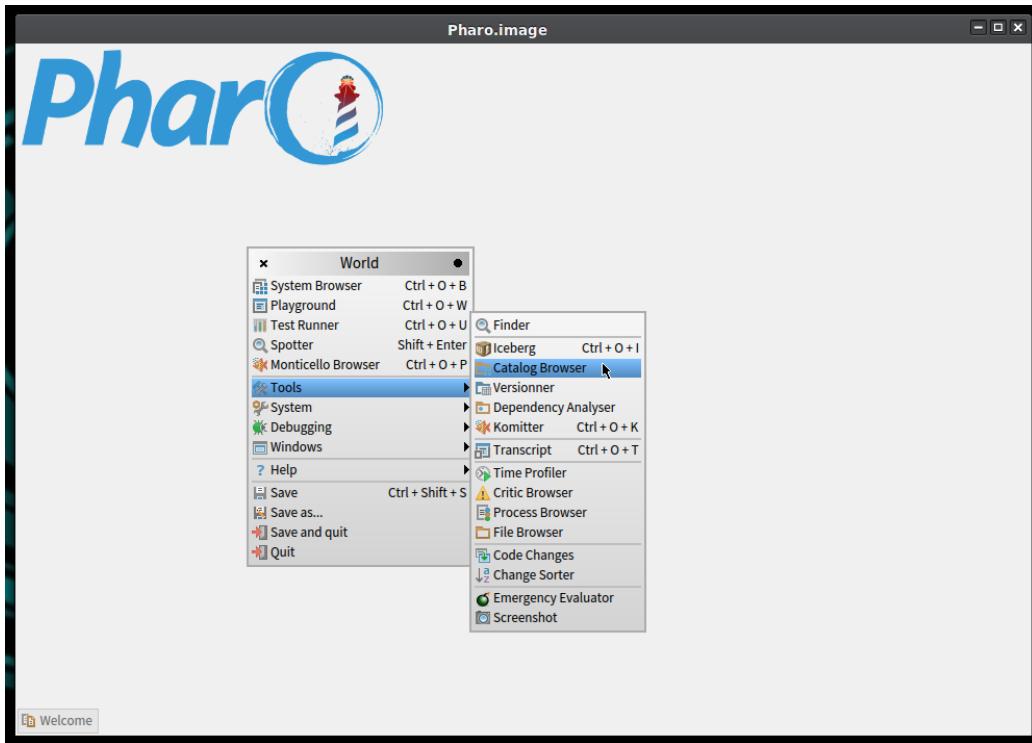
Figure 2: Opening the Cataog Browser from the World Menu.

Different install procedures suit different tastes and needs and bring alternatives, so, if one doesn't work for you or your need/taste, you can try others, or just leave it like that, if your chosen method worked.

3.1   Install from the Pharo catalog

To install Grafoscopio, from Internet in Pharo 6 or 6.1, using the Pharo Catalog, follow this steps:

1. Open the World Menu by making main (right) click in any place that is not occupied by a window. and then choose `Tools > Catalog Browser`, as shown in the figure 2.
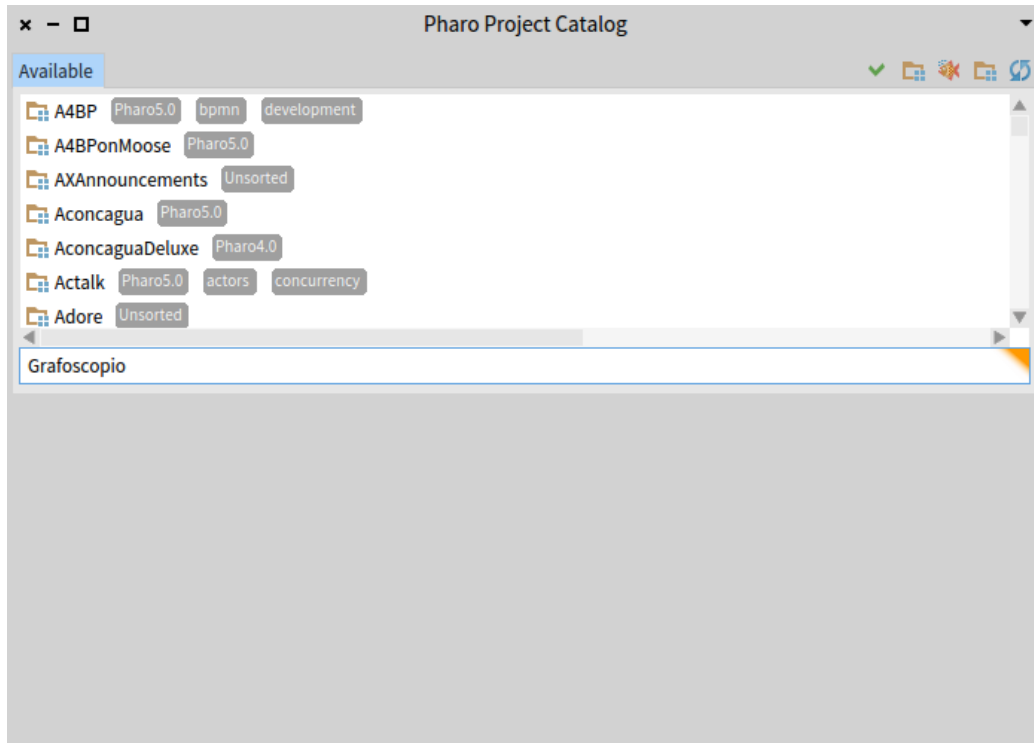
Figure 3: Searching for Grafoscopio in the Catalog Browser.

2. In the Catalog Browser window that was just opened, go to the "Search Input" field and write "Grafoscopio" (without the quotes) as shown in figure 3, an press the `Enter` key. A list of results matching the search criteria will be shown.

3. Click on the line available packages that say "Grafoscopio", it will become highlighted and the description below will be filled with details about the project/package. Then click on the most left icon above the "Available" tab, with the tooltip that says "Install stable version", as shown in figure {fig:catalog-install-stable}

4. While the installation is running, some progress bars with package names are going to be showed (see figure 5):
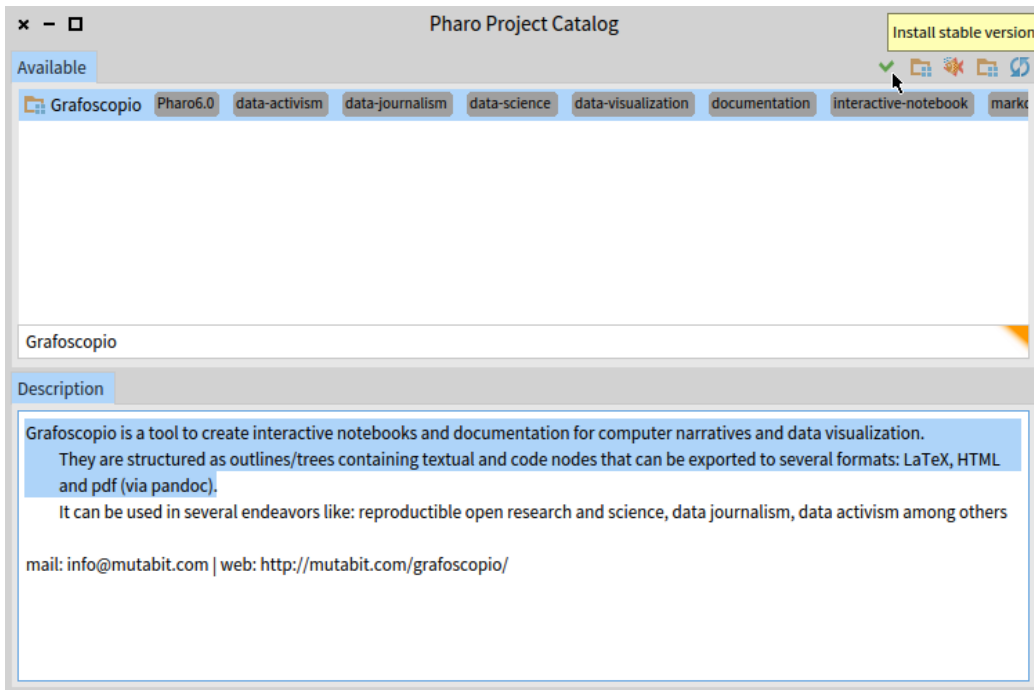
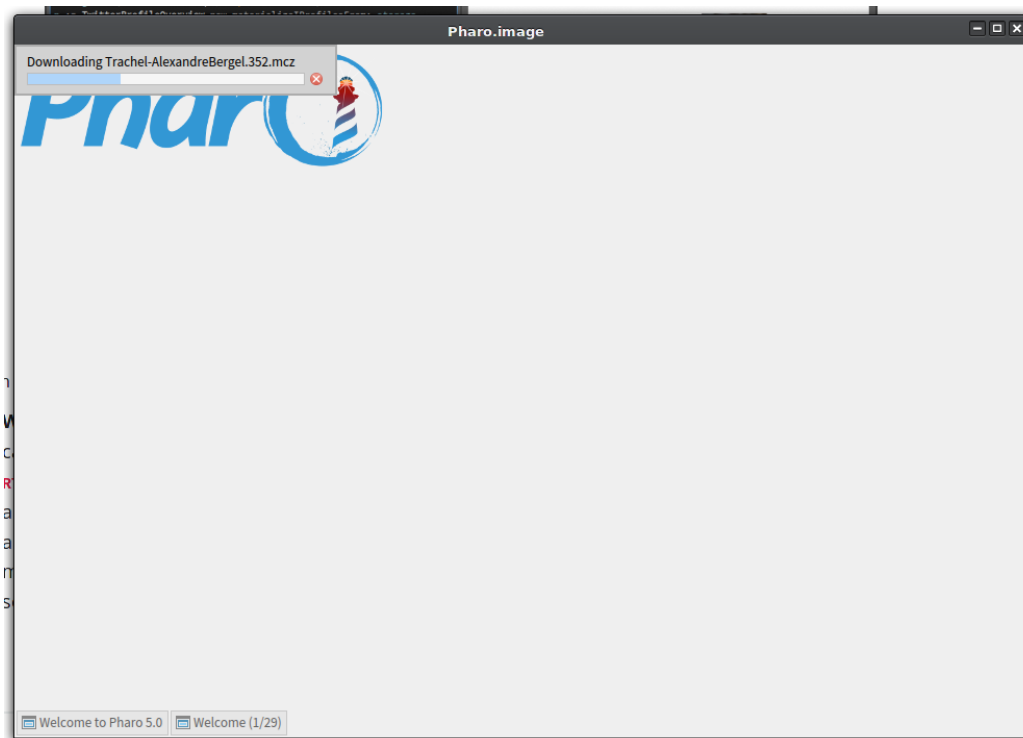Figure 4: Install stable version from the Catalog Browser.
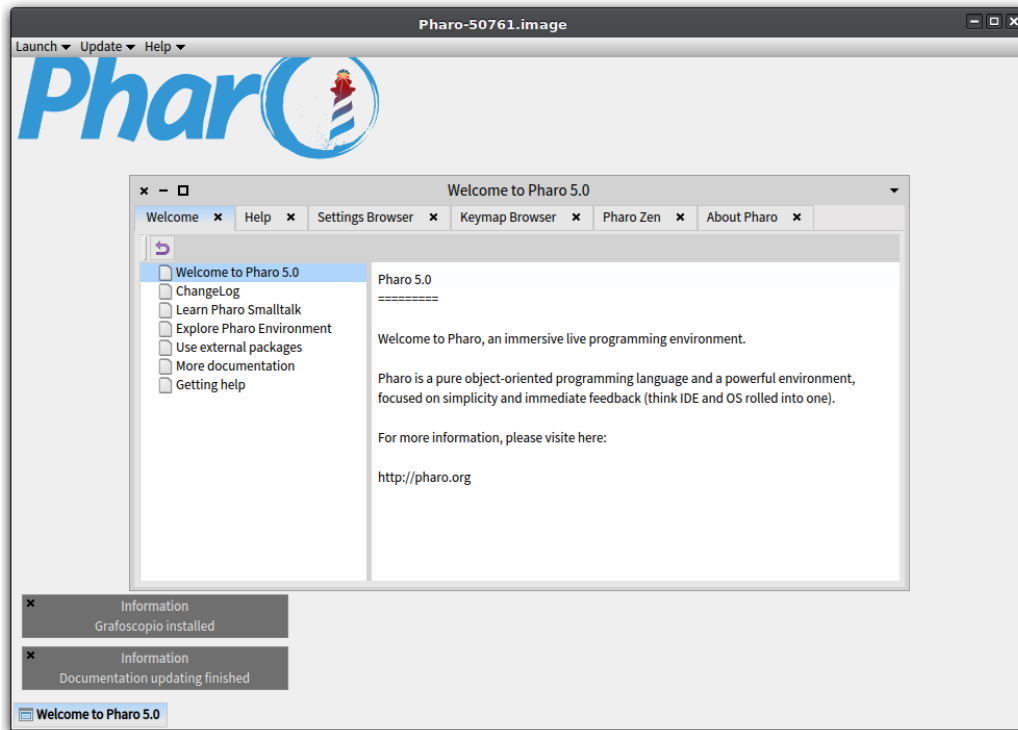
Figure 5: Installation progress bars.

Figure 6: Installation ended.

5. When the installation ends we will see two indicators (as shown in figure 6):

- Messages in the lower left corner telling that installation is complete and documentation is installed.
- A tool bar in the upper side of the Pharo window, called the docking bar.

3.2   Install from a script

There are two ways of running scripts in the Pharo environment: one by importing them from Internet and the other by writing them manually.

If you want to run a Pharo script from its web address, open the spotter (Shift + Enter) and paste the address and then press Enter to open the

Figure 7: Empty *playground* and its play button.

interactive *playground* and finally press the `Do it and go` green play button or its shorcut (`Ctrl + Shift + g`). (An empty *playground* and its play button are showed in figure 7)

Installing via scripts consists in two steps: the first step makes available the configuration of the project or package we want to install, that tells where the package/project and its dependencies are located and in which order they need to be installed; the second step runs (or loads) such configuration, making the new software available in our system. We are going to learn next how to run this two steps, by running separately from different playgrounds or by loading them progressively from the same playground.

For example, if you want to run the first part of the install script, open the spotter and paste this address http://ws.stfx.eu/ADBWX39G4FX0. You will see a screenshot similar to figure 8, showing the web address you have pasted and the first lines of the script below, marked in grey.

Press `Enter` or select with the mouse the area with the grey background. You will see the interactive playground with the script loaded. We will see more details about the playground later. For the moment press the play button or the shorcut (`Ctrl + Shift + g`). You will see that the playground has been executed. An executed playground contains a new column with details of the object resulting from that execution, as shown in figure 9.
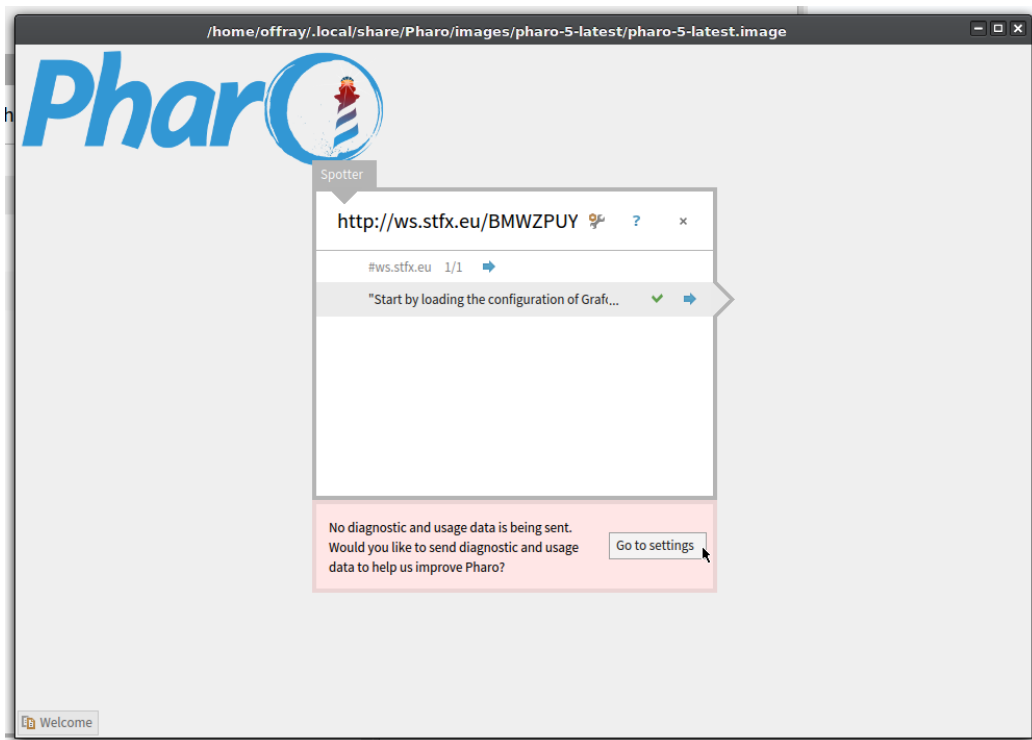
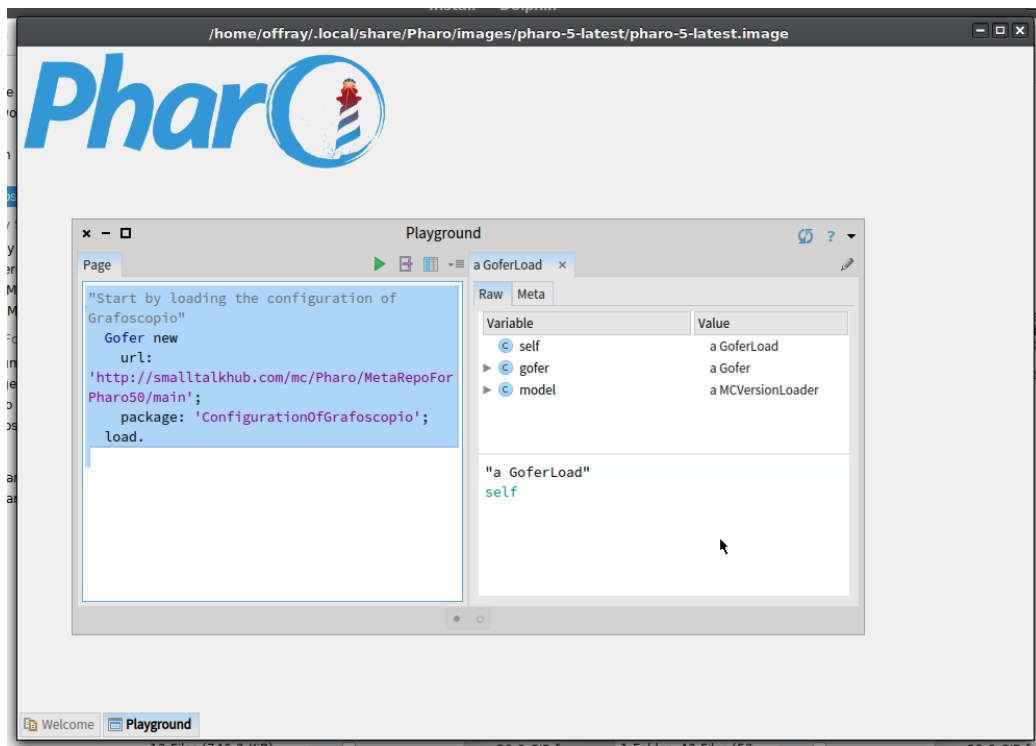Figure 8: Loading the install configuration package.
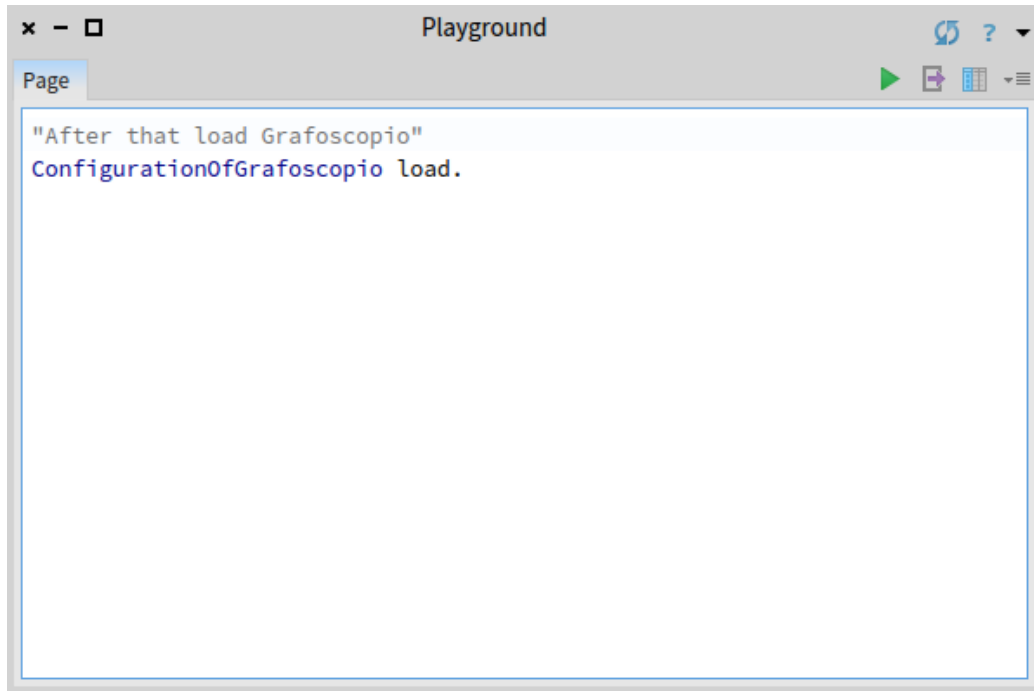
Figure 9: Executed playground.

Figure 10: Loading Grafoscopio.

Now repeat the procedure, opening the spotter, pasting this url http://ws.
stfx.eu/CZ87ZZ2SXCEM and executing the second part of the installation
script (showed in figure 10).

You will see the progress bars and the ending of the installation process, as
described in the steps 4 to 5 of the previous section.

Is usual to run the previous two steps in a single playground, by executing
parts of it. Here we are going to show you how to do it, with the same
installation example we have done so far. Open a playground (`Ctrl + o +
w`) and write this (or paste the URL of this playground, in the spotter, as
before):

```
1  "Start by loading the configuration of Grafoscopio"
2    Gofer new
3      url: 'http://smalltalkhub.com/mc/Pharo/MetaRepoForPharo60/main';
4      package: 'ConfigurationOfGrafoscopio';
5    load.
```
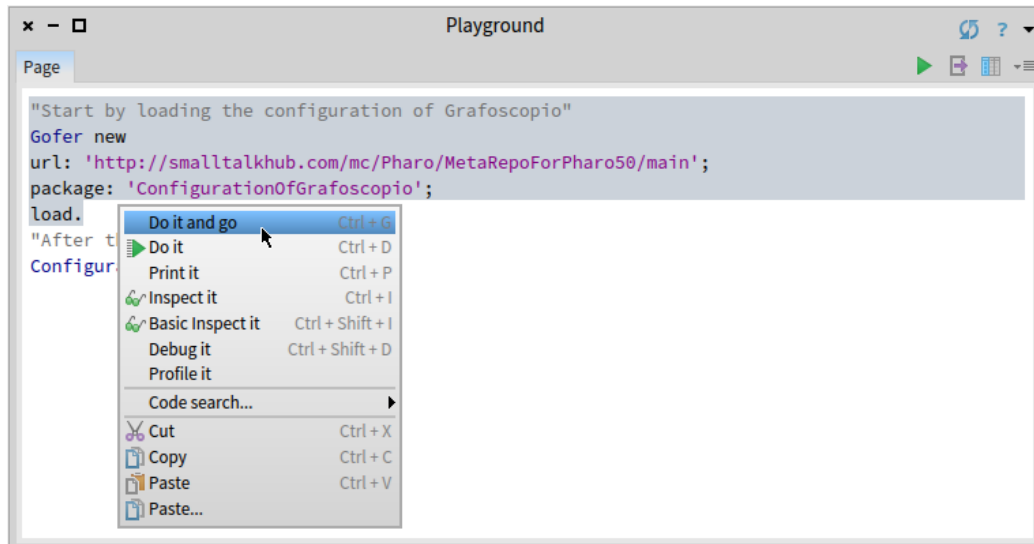
Figure 11: Selecting the script part that will be executed and deploying the contextual menu.

```
6
7   "After that load Grafoscopio"
8   ConfigurationOfGrafoscopio load.
9
```

Now select with the mouse the first 5 lines of the script and make click with the mouse secondary button. A contextual menu will be show near to the selection, as shown in the figure 11. Choose from that menu the `Do it and go` option (or press the `Ctrl + g` keyboard combination). Only the selected part of the script will be executed.

Now select the second part of the script, the last two lines, as shown in figure 12 and repeat the previous procedure. You will see the progress bars and the ending of the installation process, as described in the steps 4 to 5 of the previous section.
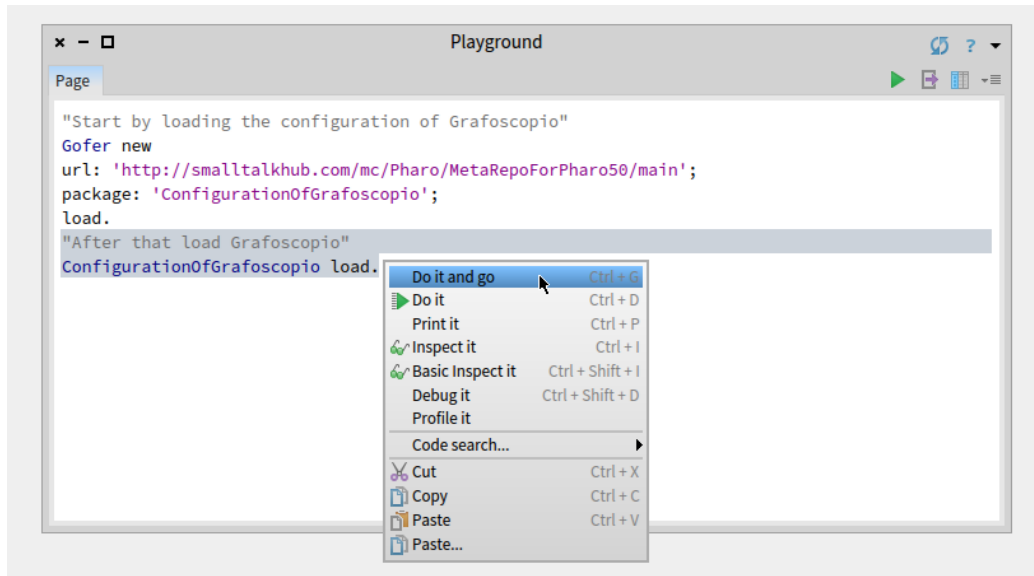
Figure 12: Executing the second part of the script.

3.3 Save the installation in your Pharo computing environment

Important: Once we have Grafoscopio installed, by any of the means shown in this section, is important to save the modifications to our computing environment, by making click in any clean part of the GUI (not occupied by any window) to deploy the *World Menu*. There we choose `Save`, to save the system with the same name, or `Save as` to save it with a new one (see figure 13).

4 Using Grafoscopio

This section will show you how to use Grafoscopio from the GUI.

4.1 The spotter

One of the first elements of the Pharo GUI that is an excellent companion for Grafoscopio is the spotter, provided by the GT Toolkit. It is a quick
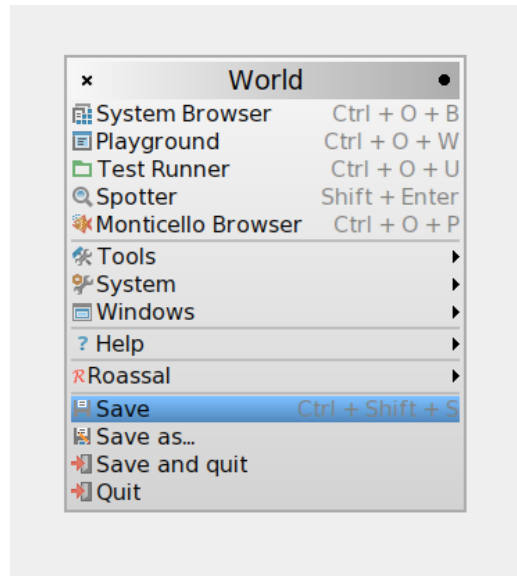
Figure 13: Saving changes to our Pharo environment via the World Menu.

finder and laucher of system functionality and behaviour, allowing you to launch and execute scripts, browse the system, and find code, among other uses. Spotter and playgrounds (used before to install Grafoscopio) are a good entry points to deep in the system. We will use them progressively and you will learn more on how to use them, as you use the interactive documentation provided with Grafoscopio and the Dataviz package.

To launch the spotter press `Shift + Enter` in Pharo. You will see something like the figure 14:

You can learn more about the spotter in the Pharo MOOC,[6] particularly in lesson 9 of Week 3 (video 1, video 2).

---

[6]MOOC stands for Massive Online Open Courseware. The Pharo MOOC is excellent and a really good entry point for programmers wating to learn more about Pharo and live coding. It also provides good complementary information if you come for other disciplines and endeavors and want to complement your reproducible research, modelling, data storytelling and visualization, with solid foundations of the Pharo environment, but it presumes familiary with programming. For an introductory book to object oriented programming, using the Pharo environment, look for the upcomming release of Learning Object-OrientedProgramming, Design and TDD with Pharo (vol I) (Ducasse 2017) (beta version here).
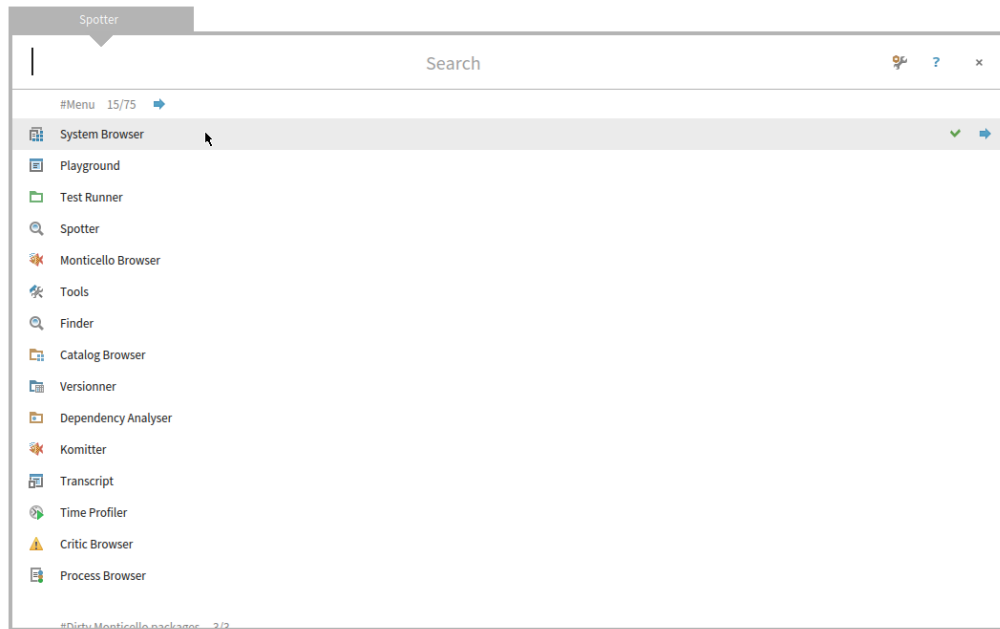
Figure 14: The spotter.

4.2   The docking bar

The *docking bar* is a fixed point in the graphical interface for quick access to certain functionalities, located at the top side of the Pharo main window see figure 15.

The docking bar is divided in three menus that are explained below.

4.2.1   Launch menu

This menu allows the creation of new notebooks and the opening of recent ones. Also it allows to launch *playgrounds* for writing scripts and code snippets and *Transcripts* to see logs of the system. Figure 16 shows the `Launch` menu.

This are the options in the Update menu:
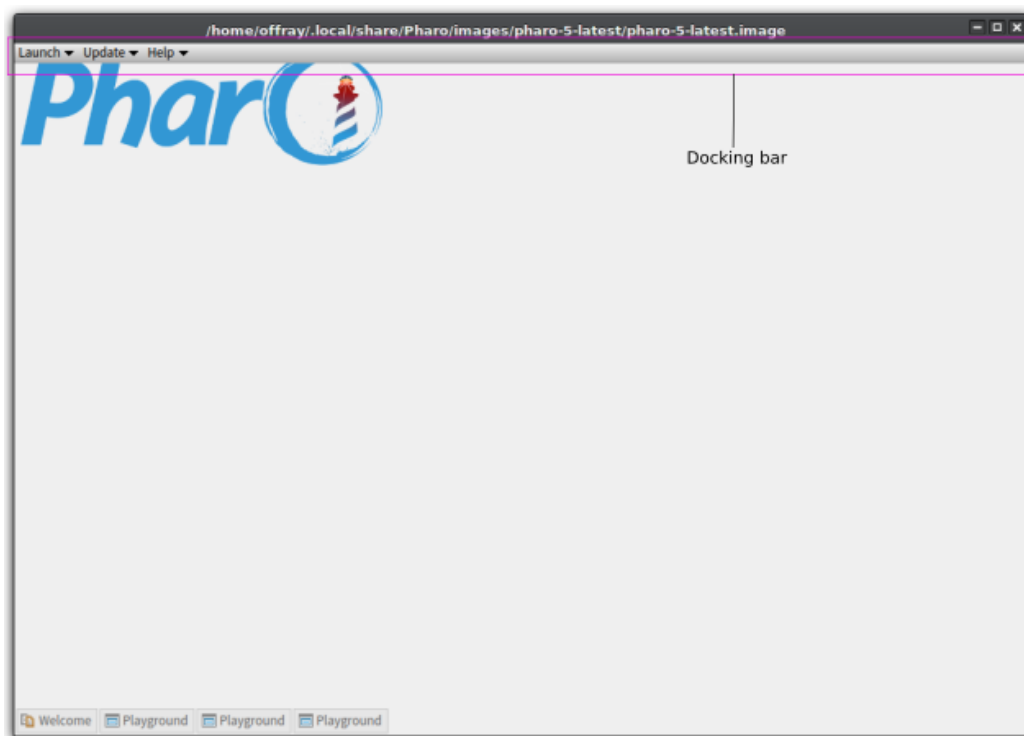
- `New notebook`: Creates a new notebook.
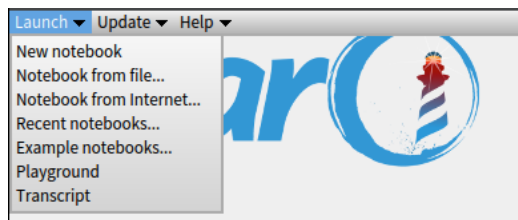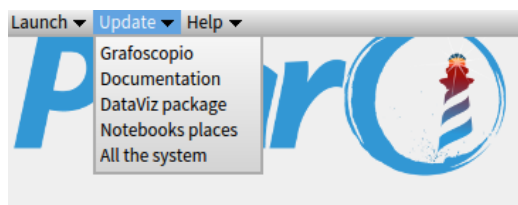
Figure 15: The docking bar.

Figure 16: The launch menu.

Figure 17: The update menu.

- `Notebook from file...`: opens a pre-existing notebook from a local file.
- `Notebook from internet...`: opens a notebook from an URL and creates a local copy in the temporal folder.
- `Recent notebooks...`: lists the notebooks that have been recently opened and/or saved.
- `Example notebooks...`: (**TBD**) Lists a set of example notebooks (may be integrated in the 'Help` menu).
- `Roassal visualizations gallery...`: Opens a visualization browser grouped by categories. Useful as starting point and inspirations for some projects.
- `Playground`: Opens an interactive coding scripting environment, with single "pages". Playgrounds are also embeddable inside Grafoscopio notebooks.
- `Transcript`: opens an output window to see log of events or see print messages.

4.2.2 Update menu

This menu updates functionality, documentation and settings for Grafoscopio. Grafoscopio has a rolling release model, so continuous updates in functionality and documentation, come after the version packaged with the default installation. The updates frequency increases with our Data Week hackathon-workshop, seminars, and other community events, so is good to go the update menu regularly. Figure 17 shows the `Update` menu.

This are the options in the Update menu:

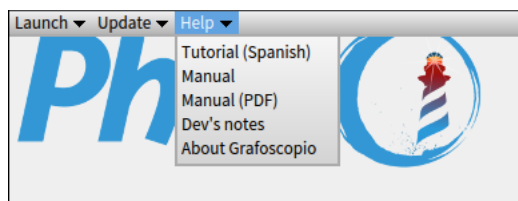- `Grafoscopio`: Updates Grafoscopio to the latest development version from the repository.

Figure 18: The help menu.

- `Documentation`: Updates companion documentation, that comes as native notebooks or in exported formats (PDF now and HTML in the future).
- `Dataviz Package`: Updates the Dataviz companion package, which contains Domain Specific Languages and Visualizations that are introduced as Grafoscopio notebooks. A companion notebook about this package is available on the Help menu, for more detailed information.
- `Notebooks places`: Updates some globally shared references for notebooks, so they can be adapted to personal preferences. This is useful for workshops, so participants can still store some notebooks in their preferred locations, while sharing relative routes. Examples of this functionality can be found in the Dataviz package notebooks documentation.
- `All the system`: Updates all of the above, except for the "Notebooks places", without going for any individual menu.

4.2.3   Help menu

The Help menu contains mostly references to interactive notebooks that teach or exemplify how to use Grafoscopio or the core technologies behind (Pharo tutorial, Roassal, STON, etc). They are installed with Grafoscopio in native format and some are in PDF, like this manual. Figure 18 shows the `Help` menu.

We have followed a local first approach for the Grafoscopio development, which implies that most of the documentation is writen for the local context first and in Spanish. The Grafoscopio Manual is the first English "formal" document (besides blog post and constant communication with the international Pharo communities that are in English). Because the GUI is in English now (it was in Spanish at the begining), we indicate when a `Help` document is

in Spanish. In the future we would like to have multilingual documentation, with a bigger team and community to work on this issues. And of course you're already invited to be part of it.

This are the options in the Help menu:

- `Tutorial (Spanish)`: Opens a tutorial used for the Data Week hackathon/workshop, that advances in increasing difficulty, covering the Pharo introductory tutorial (Prof. Steph), some simple scripts, reading of structured data (in JSON), building of a first package (Cinemania), HTML templating and programatic generation (via Mustache) and basic web publishing (via Teapot). (For a more detailed view of the contents, you're welcomed to explore the tutorial by yourself).
- `Manual`: Opens this manual as a Grafoscopio native document.
- `Manual (PDF)`: Opens this manual, exported as PDF.
- `Dev's notes`: Opens a notebook with notes from the developer. Useful to know what is happening with the development process and where is going and is complementary to other development infrastructures, like code, docs and issues repositories (provided by SmalltalkHub and Fossil).
- `About Grafoscopio`: Opens the "About" window, showing contributors and the software version.

4.3   The notebook GUI

The notebook GUI is composed of tree main parts (as showed in figure 19):

1. The top bar
2. The document tree
3. The node details

Each of these sections will be detailed below.

4.3.1   Creating a new notebook

To create a new notebook you have several options:
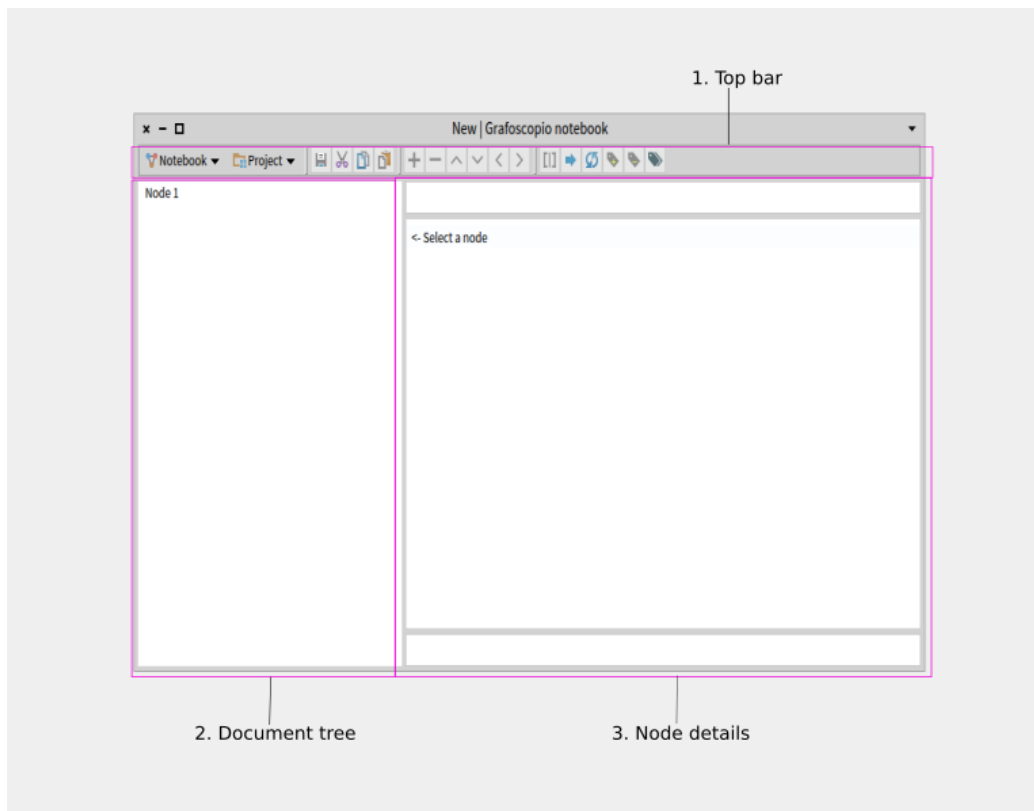
Figure 19: A new Grafoscopio notebook.

- Choose `Launch > New notebook`, from the *docking bar*.
- Open a *playground* (`Ctrl + o + w`) and write `GrafoscopioNotebook new openDefault`.

You will see a window like the shown in figure 19.

Next section will explain the functionality of the notebook GUI.

4.3.2 The top bar

The top bar groups funtionality to save, export and edit the document tree and associate it to related assets, that form projects.

This is the detailed information.

Figure 20: The notebook menu.

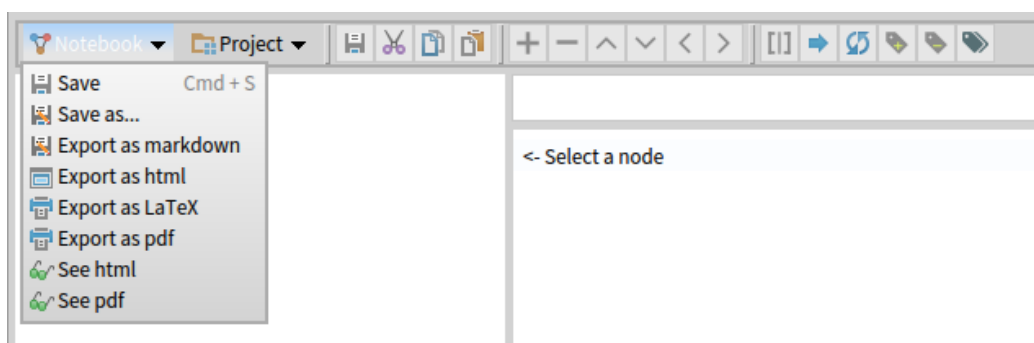### 4.3.2.1   The notebook menu

Allows the creation new notebooks and their exportation to external formats. Figure 20, shows this menu, when the notebook button is clicked.

These are the options of this menu:

- `Save`: Saves the notebook to the filesystem. If no filename has been provided, asks for one.
- `Save as...`: Saves the notebook to the filesyste under a new name.
- `Export as markdown`: Exports the current notebook to Pandoc's markdown.
- `Export as html`: Exports the current notebook to HTML format.
- `Export as LaTeX`: Exports the current notebook to LaTeX format.
- `Export as pdf`: Exports the current notebook to PDF format. Works *only* in Unix inspired systems like Mac, and Gnu/Linux.
- `See html`: (**TBD**) Shows the exported HTML document.
- `See pdf`: (**TBD**) Shows the exported PDF document.

See the "Exporting" section for details about exporting and the pandoc prerrequisites to make it work.

4.3.2.2   The project menu (TBD)

A project is a Grafoscopio notebook with the related files to produce particular outputs from it, including data files, HTML and LaTeX templates

Figure 21: The toolbar.

or other notebooks. A project tracks the historical changes on such files, allows collaboration between collective authors, exploring or unifying work variations (what is called branching and merging) and let the profreaders or audience made suggestions (via ticketing). Projects can be published on the web, providing reproducibility and increasing transparency to research claims.

Project functionality is provided in Grafoscopio thanks to the Fossil SCM (SCM stands for Software Configuration Manager and sometimes is called Distributed Control Version System or DVCS), a simple and self-contained software for asynchronous collaboration with a pretty small footprint (~2 Mb in size for all its functionality), available for major platforms, and providing truly distributed project management features (source file management, wiki, ticketing). Fossil is similar to Git or GitHub, but it is simpler that the former and does not hide functionality under closed source code software or promotes centralization, like the later[7].

Despite of not being integrated yet, several projects have been done using the Grafoscopio + Fossil combination. For example, Grafoscopio uses Fossil as backend for its documentation and reporting issues and uses the Fossil JSON Application Programming Interface (API) to query documention and update it to the last versions.

4.3.2.3 The toolbar

The toolbar provides a series of icons to access frequently used functionality, organized by groups. In the future this functionality will be accessible also via keyboard shorcuts. Figure 21 shows the toolbar.

Here is a more detailed explanation of the icons functionality in the toolbar.

---

[7]There are several alternatives to GitHub that provide a sleek and friendly web user interface (WUI), without complicated overheat, following the same spirit of Fossil, but based on git. Checkout Gogs, for a self hosted and easy to install git based source code management system.

4.3.2.3.1 Notebook saving and node cut, copy & paste

- : Saves the current notebook.
- : Cuts the current node of the document tree.
- : Copies the current node of the document tree to the clipboard.
- : Pastes the current node of the document tree from the clipboard.

4.3.2.3.2 Node addition, deletion and movement

This operations edit the document tree. You are invited to experiment with them in the new notebook already created, to make more sense of this operations.

- : Adds a new node to the tree after the currently selected one.
- : Removes the currently selected node.
- : Moves the currently selected node one place upward, unless is the first node.
- : Moves the currently selected node one place downward, unless is the last node.
- : Promotes the currently selected node one place, to the same level of its parent.
- : Demotes the currently selected node one place, making it child of the previous slibing.

4.3.2.3.3 Switching node body, visiting links, updating content and adding tags

There are two kinds of nodes in Grafoscopio now: text nodes and code nodes. This could change in the future, providing special nodes as particular handlers and viewers for particular content (images, audio, video, and so on). But with the simple combination of this two kinds of nodes, complete and complex interactive documents can be created already, and complemented with the dynamic and inmersive Pharo ecosystem. Text nodes are writen in

pandoc's markdown, with support to embedding images, bibliographic references and all the features provided by this simple, mature and extensible markup language. Code nodes are interactive playgrounds of Pharo code, allowing to dive into different objects, and integrated with the Roassal agile visualization toolkit.

- ⬚: switches a node from text to code and viceversa. Because literate computing, weaves code and text, emphasizing narrative and storytelling supported by data, the default nodes are textual ones.
- ➡: visits a node link, opening it in the web browser, if it is not empty (see node details for more information about node links).
- 🔄: updates node body acording to the node last link, which is useful for published playgrounds. In the future this icon will sync body contents with different type of links, including those pointing to etherpads or local and remote files.
- 🏷 🏷: the add/remove tags icons will be used to define custom tags to mark nodes (**TBD**). Custom tags can be used to define node traversal paths to create different outputs from a single notebook (in a similar way to `%keywords` as is explained after).

4.3.3   The document tree

The document tree is the place where the whole document is shown and provides a uniform metaphor to organize the work. It is at the left of the Grafosocopio Notebook GUI. This part will introduce the document tree and some vocabulary to be used with it.

The tree metaphor provides sequence and hierarchy to the document and its parts and is composed by nodes. Nodes have tree kinds of possible basic relationships: parent, child and sibling, which are represented visually for the indentation level of the nodes. We'll use this visual metaphor to introduce this relationships. The nodes indented inmediately at the right of other are called the children of the node at the left, and, converserly, the node at the left is said to be the parent of the nodes at the right. Nodes at the same indentation level are siblings, sharing the same parent. If a node has children, it will show an arrow head at the left. if the arrow is pointing right, the node

is collapsed meaning is not showing its children. If the arrow is pointing down, the node is expanded, meaning that is shown them. (see figure below).

Finally, If a node A is parent of a node B, which is parent of a node C, then A is said to be an ancestor of C and C is said to be a sucessor of A.

4.3.4   Node details

As we said, documents are composed by nodes organized in trees. The node has tree parts: header, body and links. We will explain the detailed composition of a node below.

4.3.4.1   Node header

The node header is a short text (one line) that summarizes the purpose or content of the node. The header is shown in two places, in the document tree, for navigation, and at the top of the node details, as a text field, for edition. When you change the edition field, the document tree will be updated consecuently in an automatic fashion.

4.3.4.2   Node body

The node body contains more extense information for the node and can take several lines of text or code. Node body is located inmediatly below the node header text input. Node body is saved automatically with each keystroke, and takes advantage of the build in persistance for all Pharo objects (called image persistance), but if you want to update the notebook contents to the file system you will need to save the notebook explicitely, by pressing the save button in the top bar or chosing the save option in the `Notebook menu`.

4.3.4.3   Node links

Node links store links to web references. Sometimes when you're making quick outlines, you need to organice external web references and have an easy way to visit them. That's the primary functionality for node links. If a node contains a valid link URL in the node links, you can visit them, by pressing the blue arrow icon ➡, which will open the default web browser in

such link. You can also use the reload icon to load the content of the URL as the body of the notebook. If the URL is a shared Playground, the content will become automatically an interactive code node.

To add links to your node, just edit the links section in them. Once the node links has been added or edited a color mark will be shown near the upper right corner of the text field box. This is known as a dirty marker. To update the header in the document tree press `Ctrl + s` (in Windows or Linux) or `Cmd + s` (Mac).

In the future node links will behave in smarter ways, storing, for example, the history of shared playgrounds for code nodes, paths to the file system to export subtrees of a document. Web browsing and content uploading are just the begining.

4.3.5 More on code nodes

Code nodes are at the core experience of the interactive documentation, exploratory computing, reproducible research and literate computing. Code nodes are full Pharo interactive playgrounds embedded inside the Grafoscopio notebook, with full functionality, including interactive inspectors, to dive into the objects, including graphical capabilities. We use extensively code nodes in our Data Week workshops, in conjunction with the integrated Monticello DVCS, to share code in agile fashion and prototype ideas quickly. Here is a more detailed view of how to use code nodes.

4.3.5.1 Executing

Once a node has been defined as code (by pressing the button), you will see a playground page inside the node body, with custom buttons for its functionality, as seen in figure 22.

This code node is a fully functional Pharo playground. You can write Pharo code there and execute it totally or partially, as we have seen already on the Grafoscopio script installing method. If you used the Pharo catalog method, take a short look on how to execute playgrounds partially or totally and the keyboard shorcuts.

Just to remember and practice, create a new code node and write this:
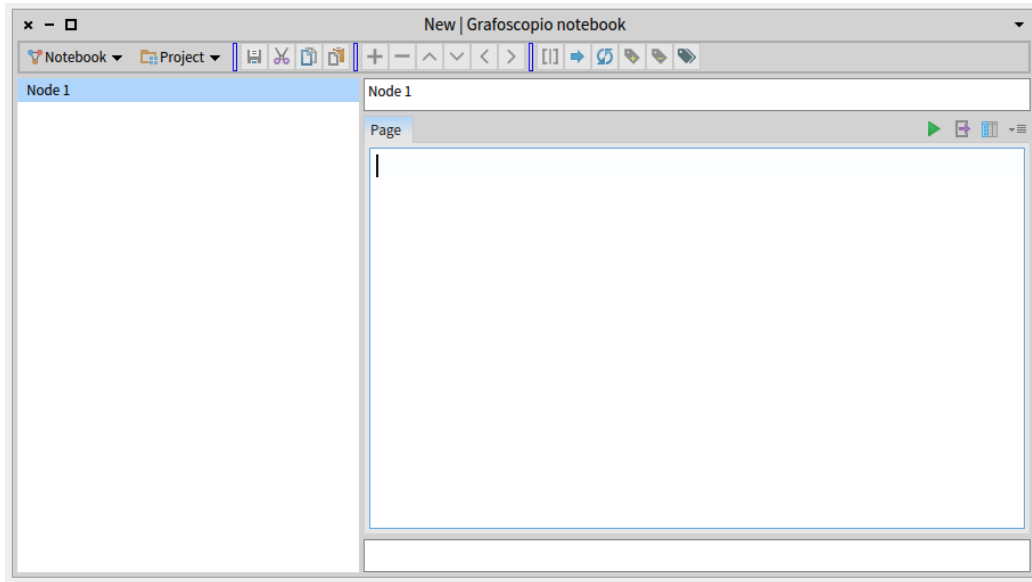
Figure 22: An empty code node.

```
1  b := RTMondrian new.
2  b shape label.
3  b nodes: (1 to: 100).
4  b edges connectFrom: [ :i | i // 2 ].
5  b layout cluster.
6  b
```

Then press the green play button ▶ (or its shorcut `Ctrl + g`) and you will see something like the figure 23:

What we have done is a small visualization (taken from the Agile Visualization quick start) that connects each number from 1 to 100 with the floor of its half (so, the numbers 18 and 19 are connected to 9, for example), and we put this visualization as an interactive code node of our notebook. You can find more examples of interactive notebooks in the `Help` menu and create your own on several topics.

4.3.5.2 Sharing
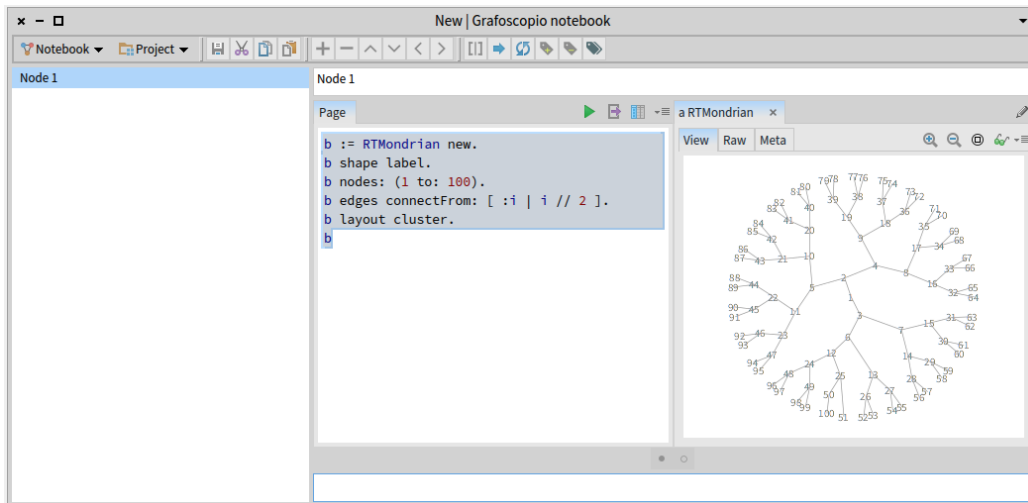
Figure 23: Executed code node.
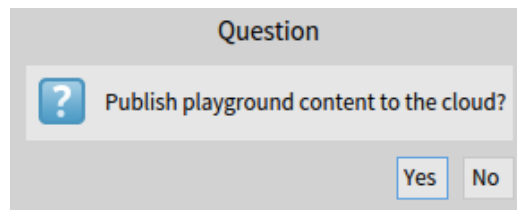
Figure 24: Publishing confirmation question.

To share a code node, click the "publish" button in the playground page [icon], that is just at the right of the play button. A question window will be open asking to confirm your publication (see figure 24).

Once you accept to share your playround contents, a notification popup will be shown, telling the web address (URL) of your published playground and will be automatically copied to the system's clipboard (see figure 25), so it can be easily pasted in mails and other messages.

Playground publications hosting services is generously provided by Sven Van Caekenberghe (also, he is the author of the superb STON format and Neo-JSON importers, used in Grafoscopio and the Dataviz companion package). It is something like a pastebin for Pharo code and a pretty convenient way of sharing small snippets of functionality in Grafoscopio. We use it continuously
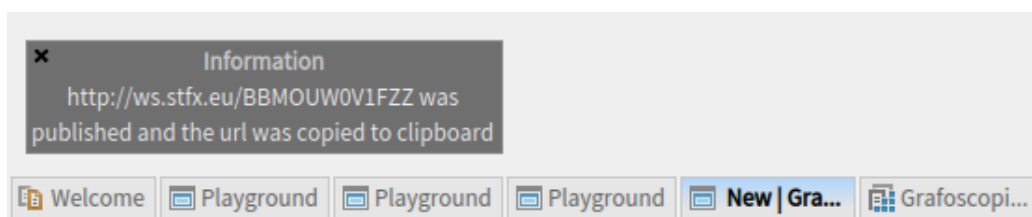
Figure 25: Publishing notification in the bottom left corner of the Pharo window.

on our Data Week hackathon/workshops.

4.3.5.3   Importing

The previous topic show you how to share code nodes from the notebook to the web. This one shows how to import them from the web to your notebooks. For this, the only thing you need to do is to paste the web URL for published playground in the node links section and press enter. The node will become a playground with the contents imported from the web.

We use this feature extensively for our workshops/hackathons and in fact was inspired by them. Usually we have a set of etherpads to write quick notes collaboratively in real time by the event participants and we put there the links of the shared playgrounds, to be imported in Grafoscopio notebooks.

In the future the node links could show the history of the shared playgrounds and its transformations.

4.3.5.4   Diving on results

Code nodes are fully functional playgrounds, as we have said. So you can use the full functionality of them to dive into the objects resulting from a code execution. By executing the playground, you not only see the results, but also a customized inspector (provided by the GT-Tools) that allows to see the attributes of the object or query them.

Several customized tabs are shown according to the execution in the playground and can be extended and adapted further.

Some of the most used tabs are:
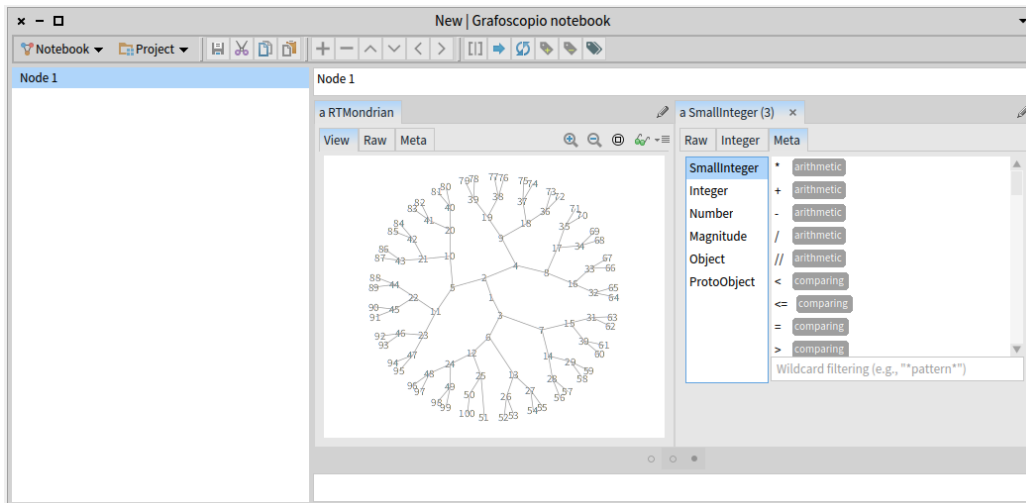
Figure 26: Diving on code node results using the playground inspector.

- `Raw`: Shows the attributes of the object resulting from the playgrounds execution. Those attributes are also objects, so you can dive into them. Once you have selected an object, a faceted browsing will start opening a new set of tabs zooming into this particular object. The lower panel of the raw view, can be used to send messages to the main object, or the objects that compose it.
- `Meta`: Shows the objects hierarchy starting with the one resulting from the playgrounds execution and ascending in the containing classes. Is pretty useful to have a quick overview of which is the vocabulary of the object, wich messages can be send and how they're organized (protocols) and implemented.

For example, click on any number in the last visualization we have been using in our code node. A new tab will be open, showing more details of that number. Then click on the `Meta` tab and select `SmallInteger`. You will see something like the figure 26.

To know more about playgrounds and how they can be used and extended visit the GT-Tools page.

4.3.6 Extending the markup: `%keywords`

Grafoscopio traverses the notebook's document tree to export it to Pandoc's markdown. Once it is in this format you can customize the output using the rich Pandoc options, adding templates, changing bibliographic citation styles, using metadata and fine tuning the result to get the PDF or HTML you want or need.

But sometimes, you want more control while writing and exporting the document. That's the role of `%keywords`, that let you extend the writing and exporting syntax to suit your needs as an author. They are programmed entirely on Pharo, and you can see how they are coded and use them as templates of your own special keywords (a topic we will cover in specialized notebooks) and in the future a full Grafoscopio document API and a DSL will be provided to create even easier customizations of Grafoscopio behavior and markup.

A `%keyword` tells Grafoscopio how to process the document while exporting and is usually located in the node's header, as the first word of it. Sometimes the `%keyword` also appears in the node's body, as we will see. Everytime Grafoscopio finds a `%keyword` in a node will process that node accordingly to a set of predefined rules.

In this part of the document, we will show you how to use the `%keywords` already present. You're invited to explore the Grafoscopio native version of this manual (in the `Help > Manual` *docking bar* menu) to see examples of how they are used in this very document.

- `%invisible`: Use it to mark a node that you don't want to become part of the exported document. This applies to that node and its children.

- `%idea`: Use it when you don't want the header of the node to become part of the exported result. Is useful to split long text that are part of a single section into a set of main ideas.

- `%embed`: Use it to change the way that children nodes are included in the exported result. If you do not use this keyword, all non-invisible children nodes will be added secuencially after the parent. If you use this keyword, all non-invisible children nodes that start with `%embed` will be added in the order they where called in the parent's body.
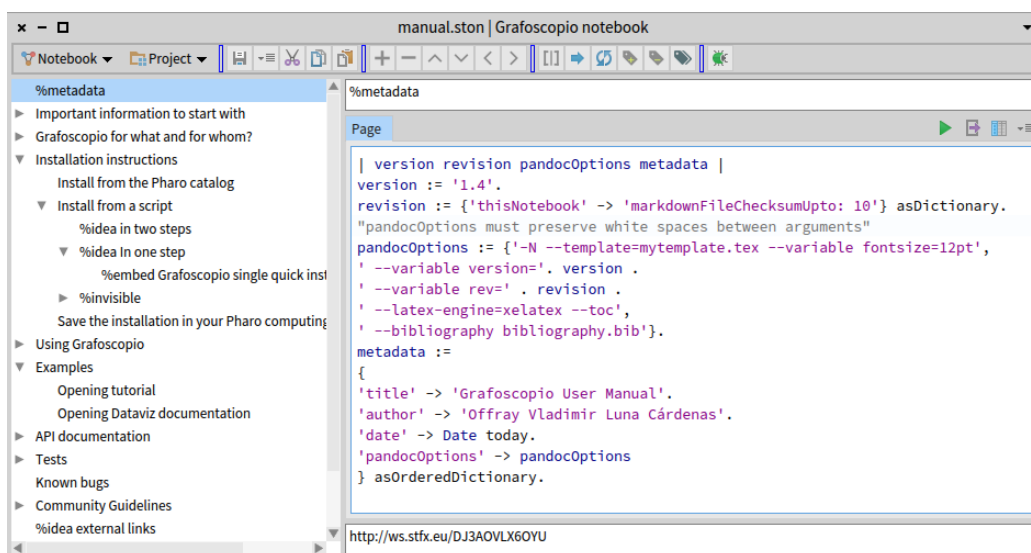
Figure 27: A metadata node example, taken from this very manual.

- **%footnote**: Use it to create a children node that contains footnotes invoked in the parents body. The header of the footnote should be the same that is referenced in the parent. So if the parents body includes a `[^my-ID]` markup reference to a footnode, the children node should have this header: **%footnode my-ID**. Useful for long footnotes, (wich are usual in academic writing) or just to have a more organized document tree.

- **%output**: Use it if you want to include the textual output of a code node execution in the exported markdown file.

- **%metadata**: Use it to mark a code node ot indicate that contains metadata about the document, that can be used to tune the exportation process to several formats. For example, when exported to Markdown, metadata nodes become the Pandoc's YAML metada block and when exported to PDF, they can control all Pandoc command options , like font size, templates, styles, table of contents, and so on. See figure 27 or open this very manual from the `Help > Manual` docking bar menu to see an example of how this is implemented.

  Metadata nodes are a powerful but intermediate level feature. What comes next asumes that you're familiar with some Pharo terminology,

if that's not the case, you can skip the remaning of this item and use the provided `%metadata` nodes in the `Help` menu and other Grafoscopio notebooks as a template for your own customizations.

Metadata is stored in the node body as a Pharo dictionary. You can store there any valid pair of key and value YAML metadata that is accepted by Pandoc, using the same pair as key and values of the Pharo dictionary. We have reserved a special key in the dictionary to give authors more fine control over the the way that their notebooks are processed. The `pandocOptions` key will have a a dynamic array as value to Indicate the command line options to be run by the Pandoc command. The dynamic array must include white spaces in the options, as expected by Pandoc, and may include variables and dictionaries. In the case of dictionaries, Grafoscopio will look again for special keywords and process the value of such keywords as a message send to particular objects. The objects that these keywords represent are:

- **thisNotebook**: Represents the current Grafoscopio notebook that is being read.
- **thisNode**: (**TBD**) Represents the node where the code appears.

4.3.7 Exporting: markdown, HTML, LaTeXand PDF

Grafoscopio provides exportation capabilities to Pandoc's markdown and use Pandoc to export to HTML, LaTeX and PDF[8]. To export a document, choose the appropiate option from the `Notebook` menu. Exporting to markdown will save the document in the same location as the current notebook, but with `.markdown` extension, instead of `.ston` and exporting as PDF, LaTeXor HTML will run the proper Pandoc command on this document for such conversion and will add the `.pdf`, `.tex` or `.html` respective file extension (provided that Pandoc is installed on your system at the usual location, if is not, Grafoscopio will send you an error message). If you want to export

---

[8]Exporting to PDF, requires the proper instalation of a LaTeX engine, like pdflatex, luatex, xelatex or context. Explanation about how to install them for each specific platfom (Windows, Gnu/Linux and/or Mac) goes beyond the objectives of this manual and is available online. Please consult the published documentation for your particular platform.

In the future we hope to provide integration with external package managers, like Nix (for Gnu/Linux and Mac) or Chocolatey (for Windows) to help with external tools and frameworks installations, in case you don't want to make them by yourself.

a document with a different name, you wil need first to save the native document with such new name.

You can also use Pandoc over the markdown file to create beautiful documents for the printing and the web, with templates and customized control over a lot of variables (see the Pandoc website for more information). Because the idea is to have reproducible documents, the best practice is to put the Pandoc exporting command in a notebook node using the `%metadata` keyword, as described before, so other readers/explorers can see how the particular PDF/HTML file was produced and using wich options. For example, the PDF version of this document was created in this way, using the `%metadata` publised in this playground link http://ws.stfx.eu/DJ3AOVLX6OYU

Important note > *Bibliographic support: Zotero and BibTeX* | Grafoscopio has preliminar support for reading and generating BibTeX files and to connect with public Zotero collections. In fact, the bibliography file for this manual was done by reading a Zotero collection and creating the respective `bibliography.bib` from there. This allows to work collaboratively on research, sharing bibliographic references, making annotations and customizing their integration into the final exported document. Open this manual from the `Help > Manual` docking bar menu and open the `Bibliography` node to see an example of how this is done for this document.

You can browse the proper folder in the fossil repository to see the manual and companion files in different formats.

5   Examples

There is a dedicated complementary package, called `Dataviz`, with examples, that was build with educative purposes, for a recurrent local workshop called the Data Week, where we invite a diverse community in gender, professions, educational and life backgrounds. The Dataviz package is automatically installed when you install Grafoscopio, and you can find more information about it in the [Dataviz intro][dataviz-intro] repository page. Also we have a Spanish introductory tutorial, that is directed towards beginers.

The notebooks for such examples are available in the `Help` menu of the docking bar, or you can open them by executing this code form the *playground*:

```
1  "This opens the Spanish tutorial"
2  GrafoscopioNotebook open: GrafoscopioDocs tutorial
```

```
1  "This opens the introductory notebook to the Dataviz package"
2  GrafoscopioNotebook open: DatavizDocs introNotebook
```

6 API documentation

Because Grafoscopio inhabits a full live coding environment, it follows the custom of making the API documentation available inside a dynamic environment, instead in some static web page. To open the Grafoscopio package inside the system browser, and see the messages organized by protocols execute:

```
1  "Browser the notebook API"
2  GrafoscopioNotebook browse.
3
4  "Browse the document tree API"
5  GrafoscopioNode browse.
```

7 Tests

The core functionality of Grafoscopio document tree model is tested. If you want to see and run the tests, just open the tests package by executing:

```
1  GrafoscopioNodeTest browse
```

From there, you can use the extensive test environment provided by Pharo. For a short introduction to how to run test on this environment, I recommend to watch the Coding a Little Domain Specific Language video, from the Pharo MOOC.
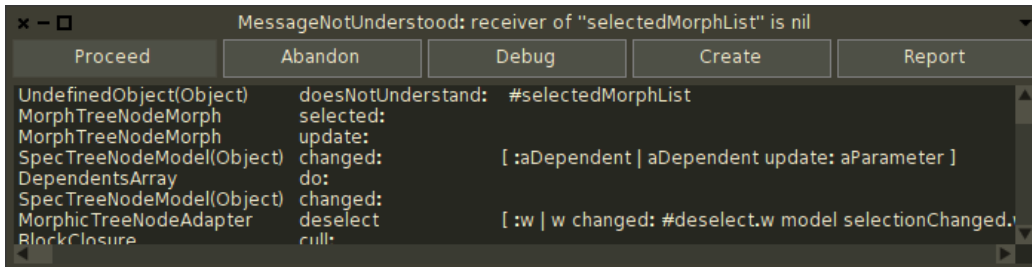
Figure 28: The know bug message.

8   Known bugs

There is a non critical but annoying bug that presents from time to time when you are using the notebook GUI.

Some times, when you click the document tree a window popup with an error message, titled "`MessageNotUnderstood: receiver of "selectedMorphList" is nil`", as shown in figure 28.

If that is the case, you still can continue your writing in the current document, clicking on other notebook nodes and editing them, but if the message presents again (usually when selecting the same node that originated it the first time), you can save the notebook and reopen it again from the `Launch > Recent notebooks...` docking bar menu.

We are going to hunt and squeeze that bug out of existance. Resistance is futile. To help us with this or other bugs please look at the Community Guidelines to know how to contribute to the project.

9   Community Guidelines

9.1   Seek support

Grafoscopio has a small and new born community. You can reach it by following the contact links in the Grafoscopio page in Spanish or in English.

Also you can discuss issues related with Grafoscopio in the Pharo users community mailing list. We are in such list and try to be active participants there and bridge the local Spanish community with the international one.

9.2   Report issues or problems

To report issues or problems with the software and/or its documentation please visit our ticket section Fossil repository. Before creating a new ticket, please be sure to visit the current tickets, to see if your issue/problem has been not reported already.

When you are filling out the issues form, take into account to include the software version where you found this problem. If you are reporting an issue with documentation, add the document version, usually found in the lower left side of the page, including the document revision in the square brackets.

9.3   Contribute to the project

As we said, Grafoscopio wants to help in blurring the distinction between software developer and interactive document author, so we are pretty open to several ways of contribution: from bug reports, as explained above, to the creation of interactive documentation, domain specific languages (DSLs) and visualizations, or software functionality.

Contributions usually take part on our recurrent Data Week hackathon/workshop and there you will learn how to use and adapt the software, starting by the basics, creating DSLs and crafting visualizations and integrating them into interactive notebooks. You will also learn how to use Fossil and how to commit to our shared repositories for code and for documents and issues. Besides this manual, we are creating also a tutorial (in Spanish) with all these themes covered, as memories for us and others to remember and learn from. The idea, as was said before, is to have multilingual documentation with a *local first* approach.

If you don't have the chance to assist to one of our face to face learning workshops and hackathons or to use the resulting notebooks, but still want and already know who to contribute, you can also ask for permisions in the respositories using any of the contact methods listed above. We are a small, new born and friendly community with low traffic mail communication and can discuss about contributions on an individual case by case approach, so your words, bugfix and suggestions will be listened and taking into account and integrated when and where they make sense.

Welcome again to our community :-).

10 Licenses

10.1 Grafoscopio, Dataviz and Pubiblio Software License

The Grafoscopio, and its companion packages Dataviz and Publibio, are software covered under MIT License

Copyright (c) 2014-2017 Offray Vladimir Luna Cárdenas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.2 Documention License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS COPYFARLEFT PUBLIC LICENSE ("LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND ALL OTHER APPLICABLE LAWS. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED IN THIS LICENSE, YOU AGREE TO BE BOUND BY THE

TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN AS CONSIDERATION FOR ACCEPTING THE TERMS AND CONDITIONS OF THIS LICENSE AND FOR AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS LICENSE.

THE GRAFOSCOPIO MANUAL, THE DATAVIZ NOTEBOOK, AND THEIR DERIVATED FILES LOCATED IN THIS REPOSITORY, ARE COVERED BY THIS DOCUMENTATION LICENSE.

10.2.1   Definitions

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **"Distribute"** means to make available to the public the original and

copies of the Work or Adaptation, as appropriate, through sale, gift or any other transfer of possession or ownership.

d. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the

Licensor to exercise rights under this License despite a previous violation.

h. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

10.2.2 Fair dealing rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

10.2.3 License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Section 4(f).

10.2.4 Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be

made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(d), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(d), as requested.

b. Subject to the exception in Section 4(c), you may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

c. **You may exercise the rights granted in Section 3 for commercial purposes only if**:

d. **You are a worker-owned business or worker-owned collective; and**

ii. **all financial gain, surplus, profits and benefits produced by the business or collective are distributed among the worker-owners**

iii. Any use by a business that is privately owned and managed, and that seeks to generate profit from the labor of employees paid by salary or other wages, is not permitted under this license.

e. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service

or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and, (iv) consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(d) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

f. For the avoidance of doubt:

g. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b) and otherwise waives the right to collect royalties through any statutory or

compulsory licensing scheme; and, iii.Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b).

g. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

10.2.5   Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

10.2.6   Limitation on liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

10.2.7   Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

10.2.8   Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Bibliography

Bergel, Alexandre. 2016. *Agile Visualization.* LULU Press.

Black, Andrew, Stéphane Ducasse, Oscar Nierstrasz, Damien Pollet, Damien

Cassou, Marcus Denker, Dmitri Zagidulin, Nicolai Hess, and Dimitris Chloupis. 2016. *Pharo by Example.* Square Bracket Associates.

Ducasse, Stéphane. 2017. *Learning Object-Oriented Programming, Design and TDD with Pharo (Vol I).*

Eghbal, Nadia. 2016. "What Success Really Looks Like in Open Source."

Girba, Tudor, Andrei Chis, and Oscar Niertrasz. 2014. "SCG: The 'Moldable Debugger'."

Hill, Benjamin Mako. 2013. "'When Free Software Isn't Better' Talk."

Luna Cárdenas, Offray Vladimir. 2014. "Metáforas Y Artefactos Alternativos de Escritura Para Jalonar La Investigación Abierta Y La Ciencia Ciudadana Y de Garage," September.

Perez, Fernando, and Brian E. Granger. 2015. "Project Jupyter: Computational Narratives as the Engine of Collaborative Data Science."