

Chapter 1

Introduction

Computers play a formidable extension of the human brain: a computer liberates us from performing boring and repetitive tasks. Data visualization is a wonderful field where computers nicely complement what the brain excels at.

Conveying information through interactive visualizations is both an art and a sophisticated engineering process. When crafting a visualization, many decisions have to be taken based on a personal intuition or a carefully evaluated design aspect. Being able to quickly experiment a new idea is key in. Agile Visualization is about leveraging creativity by reducing the cost associated to data and software visualization.

Challenges in data visualization does not actually involve visualizing data. There are numerous books and software libraries for that purpose. The challenge is in crafting a visualization that is easily reusable, composable, and extensible. Agile Visualization addresses the need to easily reuse visualizations and desire for many to reuse visualizations with their own data.

Roassal is a visualization engine, written in the Pharo and VisualWorks programming languages. All the examples provided in this book are therefore made for Roassal and are written in the Pharo programming language.

Since there is no better way than programming to craft a visualization, readers are expected to have some programming experience to fully enjoy Agile Visualization. This book is written for a large audience.

1.1 Agile Visualization

Agile Visualization promotes the creation of a visualization based on very short incremental steps. By "short" we mean a couple of seconds or minutes. A data analysis is carried out by building a number of visualizations, from which many are simply a try or have a very short usage time period. Reducing the creation time of a visualization to a few seconds or minutes greatly increases the number of different paths taken by a data scientist to solve a given problem.

A painter throws colors on a canvas as part of its activity. Colors may be mixed, removed, and added at will. This metaphor may be considered as the guiding line of Agile Visualization. Similarly to Agile Programming, feedback should always occur a short time after the inception of a visualization.

1.2 The Roassal visualization engine

Roassal is a visualization engine developed in Pharo. Roassal is also available for VisualWorks and is distributed under the MIT License.

Although Roassal regularly receives contributions from all over the world, the effort behind Roassal is leaded at the University of Chile and at Object Profile.

Roassal is developed and maintained in Pharo. Roassal is heavily maintained on the VisualWorks platform. Roassal runs also on BeeSmalltalk. The book is mostly written using Pharo: examples and screenshots are made on Pharo. VisualWorks developers will have to manually adapt the examples and instructions.

1.3 Online and free version of this book

The website <http://AgileVisualization.com> contains a free version of the book. An HTML version of each chapter is provided. Buying a physical copy of Agile Visualization helps facing the cost involving in maintaining Roassal and Agile Visualization.

1.4 The Roassal community

As with any successful open source project, Roassal is driven by an active community effort. The positive aspect of this is that Roassal is evolving every

day (literally). The negative aspect is that documentation becomes obsolete as soon as it is published. The book has been written in such a way that deep technical aspects are not discussed while general concepts are presented in depth. The reason is that concepts are stable over time.

In case you wish to discuss some aspect of Roassal, need help, or simply have a friendly chat, there is a number of ways to get in touch with us:

- The moose mailing list. <http://moosetechnology.org> is the website of the Moose analysis platform. Details about how to join the Moose-Dev mailing list are provided, in the *community* section. The community is friendly, open-minded, and happy to help.
- hello@agilevisualization.com is directed to the main developers of Roassal.
- The Roassal team is also available from the channel #Roassal on the Slack messaging. Information on joining the channel is available on <http://pharo.org/community>.
- <https://www.facebook.com/ObjectProfile> contains many examples of works carried out with Roassal.
- The Object profile Tweet account @ObjectProfile is also a great way to share projects.

1.5 Other visualization platforms

Although Roassal, Pharo, and Moose are heavily promoted in this book, our goal is not to make you switch language or visualization API. You probably have a good reason for using what you are currently using. In case you are seeking a fresh experience for visualizing and programming, then we cannot recommend more this tool trilogy more.

The data visualization community has produced many visualization engines. D3js, RaphaelJS, Processing, Flare are popular visualization frameworks supported by a large community. Agile Visualization presents a compelling use of visualizations in many different situations. We believe that even if you are not a Roassal user, Agile Visualization will remain beneficial.

1.6 If you do not know Pharo

Pharo is easy to learn and use. It comes with fantastic programming tools to make you intimately interact with *objects*; an object being an elementary

computational unit. Our advice is to resist the natural tendency to map your knowledge from Java or PHP into Pharo.

1.7 If you are already a Pharo/Smalltalk programmer

If you are a Pharo programmer with some graphical API knowledge, then you may want to work with the core of Roassal (Part I). Once you understand the basic functionalities of Roassal, we invite you to read more about the builder infrastructure (Part II). Builder enables fantastic scripting and visualization composition not seen anywhere else!

One last piece of advice before beginning your journey: do not try to build any complex visualization in Roassal without first reading the builder infrastructure. Builders may ease your life!

1.8 Improving the book and Roassal

This book is the result of many long discussions, compromise, and collaborative efforts. Roassal and Agile Visualization are evolving fast and will continue to do so. There are many ways you, the reader, can help us improve the book. Here are a list of actions one can take to contribute to this global effort:

- Let us know about your project. We will make sure that adequate advertisement will be made. All projects, ranging from a student effort to an industrial case study, are relevant.
- As any piece of software, Roassal contains bugs. Reporting bug is a great and easy way to help.
- Fixing bugs and proposing improvements.
- Buying a physical copy of this book contributes to support the engineering effort and to prepare the next edition of the book.
- It is difficult for us to imagine all the different ways people will use the artifacts promoted in this book. Actually, it would be presumptuous to think so. All feedback is welcome. Let us know what you wish to read and what you wish to read more about.
- Making a simple "Hello" on one of the multiple communication channels.

1.9 Acknowledgements

The book is the result of multiple and long-lasting collaborations. First of all, we would like to thank the Lam Research company. Lam Research's team has always been supportive both morally and financially. Thanks CH and Chris! You made this book a reality.

Some of the chapters are external contributions. We therefore thank their authors:

- Bui Thi Mai Anh, Nick Papoulias and Serge Stinckwich for the chapter *Expressing Epidemiological Models*.
- Leonel Merino for the chapter *Network Latency*.
- Yuriy Tymchuk for the chapter on *Domain-Specific Visualization Made Easy*.
- Onil Goubier and Thierry Goubier for the chapter *OpenStreetMap Integration*.
- Peter Uhnák for the chapter *Applying Layout*.

Many people within the Moose, Pharo and ESUG communities have deeply contributed to what is presented in the book. Your enthusiastic support and trust in what we do has always been invaluable.

We also would like to thank you, yes you, the reader, for your questions, support, bug fixes, contribution, and encouragement.

We are deeply grateful to for their contributions to (no particular order) CH Huang, Chris Thorgrimsson, Tudor Gîrba, Renato Cerro, Stéphane Ducasse, Yuriy Tymchuk, Natalia Tymchuk, Juraj Kubelka, Juan Pablo Sandoval Alcocer, Milton Mamani, Vanessa Peña, Ronie Saldago, Alvaro Jose Peralta, Pablo Estefo, Igor Stasenko, Faviola Molina, Ricardo Jacas, Daniel Aviv Notario, Sergio Maass, Serge Stinckwich, Bui Thi Mai Anh, Nick Papoulias, Johan Fabry, Nicolai Hess, Miguel Campusano, Peter Uhnák, Martin Dias, Jan Blizničenko, Samir Saleh, Nicolai Hess, Leonel Merino, Volker, Pierre Chanson, Andrei Chis, Thomas Brodt, Mathieu Dehouck, Miguel Campusano, Onil Goubier, Thierry Goubier, Esteban Maringolo, Alejandro Infante, Philippe Back, Stefan Reichhart, Ronie Salgado, Steffen Märcker.

Part I

Roassal

