Lua Programming Gems

Lua Programming Gems

edited by
Luiz Henrique de Figueiredo
Waldemar Celes
Roberto Ierusalimschy

Lua.org

Rio de Janeiro
2008

Lua Programming Gems
edited by Luiz Henrique de Figueiredo, Waldemar Celes, Roberto Ierusalimschy.

ISBN 978-85-903798-4-3.

Book cover by Pedro de Mazza Cerqueira. Lua logo design by Alexandre Nako. Typesetting by the editors using LaTeX.

Book web site: `http://www.lua.org/gems/`

Contents

Preface

It gives us great pleasure to publish this collection of Lua gems. Not only does it record some of the existing wisdom and practice on how to program well in Lua, but it also reflects the maturity of the Lua community. It is gratifying to see that Lua has motivated other people to learn it well and to share their knowledge with other users. In well-written articles that go much beyond the brief informal exchange of tips in the mailing list or the wiki, the authors share their mastery of all aspects of Lua programming, elementary and advanced.

Producing this book has required several steps. In response to a call for contributions, we received over 70 abstracts, selected 43, and received full versions for 28 of these. The authors received our comments and suggestions to prepare the final version of their articles. The whole process took two years, much longer than we had imagined. The selection of abstracts proved to be surprisingly difficult. Many potentially good submissions could not be accepted due to space limitations. Despite the long time it took and the amount of work it required (or because of it!), we are very happy to have this collection of articles on Lua contributed by members of our community. We trust the book was worth waiting for.

We thank all the authors for their hard work on the articles and everyone that submitted abstracts in the first phase. We also thank the whole Lua community for its friendliness and expertise. The active participation of our users has been to us a constant source of motivation for improving Lua. Finally, we give our warm thanks to Cameron Laird and Mark Hamburg for writing forewords to this book.

Additional material and errata will appear in the book web site:

`http://www.lua.org/gems/`

The Lua team
Rio de Janeiro, November 2008

Foreword

by Cameron Laird

When I need a programming language that's as easy as possible to embed, I choose Lua. Lua isn't just supple, free, portable, and compact, though; it's also powerful — and to get the most out of it, I'm glad I have *Lua Programming Gems*.

I need to explain that I mean something specific by that. Most of my reading is on the 'Net: I look up references, I read tutorials for unfamiliar material, I moderate a half-dozen Wikis, and I chat about specific techniques with colleagues on-line. My consumption of books has nose-dived. *Lua Programming Gems* is a book worth reading, though: its individual chapters get across ideas that simply aren't explained anywhere else.

Lua Programming Gems emphasizes practicality in a way I like. While the six authors in Part III certainly employ classroom concepts correctly in their examinations of "Algorithms and Data Structures", they do it all with working Lua code. The same pattern is apparent throughout *Lua Programming Gems*: it's filled with ideas likely to help me in my next programming project.

If you're new to Lua, you might be anxious about what you'll find. You can see that Lua offers definite advantages, but how hard is it to pick up what's undeniably a minority language? *Lua Programming Gems* will ease your concerns: the authors write clearly, modestly, and even deftly. The very first chapter, for example, tackles difficult material, including dynamically-allocated per-thread storage. The tone is consistent throughout the book. Rather than show off their expertise or indulge in private jokes, habits common for authors from other domains, the *Lua Programming Gems* authors focus on the specific details and examples that best teach their chosen topics. They make it inviting to dig deeper in Lua than you might do on your own.

Among the highlights of *Lua Programming Gems* for me: Part IV gives insight into "Game Programming", an area where I'll probably never work, although many of the techniques apply more broadly; Part V on "Embedding and Extending" is crucial for much of the programming I like; and Chapter 13, "Exceptions in Lua", is a particular interest of mine.

Do you want to “program well in Lua”? The Lua team set that as a goal when it first announced its plans for *Lua Programming Gems*. The final result fulfills that goal; you’ll like it.

Lua and Lightroom

Mark Hamburg
Founder, Adobe Photoshop Lightroom

When we started work on the project that would become Adobe Photoshop Lightroom, we knew we wanted to make scriptability an important part of our story, so early on we reviewed the usual suspects. What drew us to Lua was its combination of simplicity, power, ease of embedding, and relatively high-performance. Having a straightforward license helped too when it came time to talk to Adobe's lawyers. Personally, as an old Scheme fan, I was drawn to its first-class closure support. I also found the coroutine system intriguing. The relative minimalism also resonated with a back-to-basics attitude that had us weaning ourselves away from intensive C++ usage and back toward C.

Still, it was hard to position Lua as anything other than an obscure choice. We could cite heavy use in the games community and we had set out with a mission of learning something from game developers, but if asked what materials one could turn to learn Lua or where we would find experienced Lua programmers, the answers were limited. For the former, we had the well-written reference manual, some good material on the Lua users wiki, and an intelligent forum on the Lua mailing list. This was good material, but there wasn't a lot of it. For the latter question, our answer was essentially "Any programmer worth hiring ought to be able to learn Lua quickly." This was a situation we were prepared to deal with and the arrival of *Programming in Lua* certainly helped, but it was easy to understand why it might be off putting to someone looking in from the outside.

Why this matters is that along with Lua's simplicity come some issues that make people with backgrounds in other languages stumble. The beauty of a small core is that there is a real opportunity for mastery. This is one of C's great strengths as well. That small core, however, comes at a price. For example, Lua has no syntax for exception handling. C doesn't either but having one seems almost required in modern languages. Lua has a syntax for object-oriented

message sends, but the actual implementation of an object system or class system is a roll-your-own affair in Lua. As a result, one sees such issues raised repeatedly on the Lua users list as people new to the language start using it and then ask "But what about ____ ?"

Programming in Lua provides answers to some of these questions but those answers are necessarily terse. *Lua Programming Gems* dives deeper on these issues and many more. It shows ways to deal with threading — an issue we went through a few iterations on in Lightroom — and gives extended examples of how to hook Lua into your application. You may not always like the answers. For example, the object system presented here is quite minimalist. In the spirit of Lua, however, you remain free to roll your own using or not using the ideas presented here. The value comes in seeing the well worked examples together with a discussion of motivations and comparisons to other approaches. If this book had been around during Lightroom's development, we probably would have happily adopted some of the techniques it presents while simply taking inspiration from others. As it was, we largely had to find our own way and while that was rewarding in itself, the Lua community and particularly new Lua users can be happy to now have a field guide that maps out some of the trails.

The broader lesson from Lightroom that I would like to leave Lua users with is that you should let it pervade your work. Lua is sometimes described as being a language for gluing pieces together, but as we discovered that glue can extend quite deep. We started out looking for a scripting language for a native code application. Then we started thinking it would be nice to allow Lua to exist as a peer to native code. In the end, we ended up with a system where native code provides the foundation, but it is effectively a second-class citizen in the application as a whole. Large portions of Lightroom ended up getting written in Lua including the object-relational mapping layer for the database and the layout system for views. Lua defines the structure of the application and its extensibility mechanisms. As a result, we had an application that was smaller by far than some of its competitors, easy to change, largely cross-platform in its implementation, and suffered essentially no compile-link cycle. The reason things work out this way is that Lua is both very expressive compared to most native languages and sufficiently efficient that you can let it do a lot more of the work than one might be tempted to in other scripting languages. At the same time, the boundary between native code and Lua is sufficiently clean and efficient that when we needed to do things in native code, it wasn't a huge burden to expose that functionality to Lua nor to access functionality written in Lua.

So, my advice to Lua users and potential users is to think seriously about how widely you can let Lua spread through your work, be grateful for books like this one and *Programming in Lua* and be even more grateful for the work that the Lua team has done and their generosity in sharing it with the world.

Contributors

Following the Brazilian tradition, the contributors are listed in alphabetical order of first name.

André Carregal was introduced to Lua in 1994 during his MSc in Computer Science, which was supervised by Roberto Ierusalimschy. He has been working with web development using Lua since 1996. He currently coordinates the Kepler project and the LuaForge site while working as a consultant for Lua-related projects.

Ben Sunshine-Hill is a PhD student at the University of Pennsylvania, studying computer graphics. He did his undergraduate studies at the University of Southern California and received an MSc from the University of California, Los Angeles. He has been a game developer at several mobile and mainstream game development studios, and has previously published work on real-time rendering methods.

Diego Nehab was introduced to Lua in 1996, while working for Tecgraf in PUC-Rio. Over the years, he has been involved in a variety of Lua-related projects, including the IupLua, CDLua, IMLua, and LuaSQL libraries. He is best known as the author of the LuaThreads and LuaSocket libraries. Diego received a BEng in Computer Engineering and an MSc in Programming Languages from PUC-Rio, under the supervision of Roberto Ierusalimschy. He later received an MSc and a PhD in Computer Graphics from Princeton University. His research now focuses on high-quality shape acquisition and on real-time rendering techniques.

Doug Currie develops award-winning medical devices with Sunrise Labs, Inc. in Auburn, New Hampshire, USA. Over a thirty-year career, Doug has led electronics, mechanical, and software teams developing high-tech products with particular emphasis on reliability and adaptability. Some of these products, based on a massively parallel computing architecture Doug invented, are used in national transportation and world-class manufacturing operations. With a

special interest in little languages, Doug has also contributed technically to open source projects such as Moscow ML, Hibernate, Gambit Scheme, and SICStus Prolog. Doug holds an S.B. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology.

Eduardo Ochs is a mathematician, or sort of; his interest on simplification of proofs led him to Non-Standard Analysis, and from there he drifted to Logic, Type Theory and Categorical Semantics. In parallel with his "normal" academic life he has been a contributor to the GNU Project since 1999, and his main focus areas in Free Software are little languages and programmable textual interfaces. He keeps a big, messy homepage at `http://angg.twu.net/`; all the html pages in it are generated with "BlogMe", another extensible little language built on top of Lua.

Gavin Wraith is Emeritus Reader in Mathematics, Sussex University, UK. Joined Sussex University in 1963, retired in 1999. Founding chairman of the Sussex University Computer Science department in 1985.

Jérôme Vuarand is a young software engineer specialized in AI and working in the video games industry. He discovered Lua while looking for an embeddable scripting language, just when Lua 5.1 was released, and he fell in love both for the language itself and its new package system. His initial motivation was to move away from legacy in-house script engines, but he's now using it as a general programming language in all his personal projects, from mobile robotics to modern game engines entirely written in Lua.

John Belmonte is a software engineer currently residing in New York City. He happened upon Lua as a video game developer in 2000 and was among the first to embed it into a home console title. Since then he has been active in the Lua community through chartering lua-users.org, participating in workshops, and contributing to the language's evolution.

Julio M. Fernández-Díaz has a PhD degree in Mining Engineering (1989). He is Professor of Applied Physics at the University of Oviedo in Spain and researches mainly in the field of atmospheric aerosols. His interests lie in developing physical and mathematical simulations on computers. His first 'computer' was an HP25 calculator in 1977. As a programmer, he uses Fortran, C, Lua, Tcl/Tk and Postscript, usually as part of his research.

Konstantin Sokharev professionally develops video games since 2001, successfully completed two RPG/RTS projects for PC, one for PocketPC/Palm. He currently holds a post of technical director at IceHill llc. developing Action-RTS title "Empire Above All" and several unannounced projects.

Luis Eduardo Ximenes Carvalho has a BSc (1997) in Civil Engineering from the Federal University of Ceará (UFC), an MSc (2000) in Transportation

Engineering from the Federal University of Rio de Janeiro (UFRJ), and an MSc (2002) in Computer Science from UFC, all in Brazil. He is currently a PhD candidate in the Division of Applied Math at Brown University, where his research comprises applications of Bayesian statistics to computational biology. He also has interest in logistics and optimization, scientific computing, graph theory, and programming languages, especially Lua.

Matthew Burke is an Assistant Professor of Computer Science at The George Washington University in Washington, D.C. Lua has replaced Forth as his favorite language in which to program while riding the subway, and he does so using whatever device is serving as his PDA du jour. He is also developing a curriculum for introductory Computer Science which uses Lua. When not programming, he likes to travel with his wife and son. He was the organizer of the Lua Workshop 2008.

Nicolas Peri is co-founder and technical director of the French company Stone-Trip, creator of the 3D game development platform ShiVa. He is in charge, among other things, of the ShiVa scripting engine, which is based on Lua. Before that, he worked as engine developer for other gaming companies, including Kalisto Entertainment and UbiSoft Tiwak.

Patrick Rapin studied at the Swiss Federal Institute of Technology at Lausanne (EPFL). He is now a software engineer working for Olivetti Engineering at Yverdon-les-Bains, developing printer firmware, image processing algorithms, and printer test tools.

Pedro Miller Rabinovitch, a PUC-Rio graduate, has worked with Lua at Tecgraf and Cipher Technology, and is currently a game developer at Jagex.

Rafael Moreira Savelli graduated in Computer Engineering at PUC-Rio. He worked for Tecgraf in PUC-Rio for over four years. He is now studying for an MSc at UFF and working in the Visgraf laboratory at IMPA.

Ralph Hempel is a Professional Engineer in Ontario, Canada and specializes in designing embedded systems. After learning to program on an HP41C, he never lost his fascination with small languages and hacking consumer products. He wrote pbForth for the LEGO MINDSTORMS RCX and then ported Lua to the NXT. When he's not wrangling embedded systems, Ralph enjoys mountain biking in the summer, snowboarding in the winter, and ice hockey all year long.

Ralph Steggink joined Océ in 2001. With a degree in both chemistry and computer science, he now develops controller software for printers. Together with Wim Couwenberg he prototyped revolutionary concepts using Lua. These currently find their way into several Océ products. He is an enthusiastic volleyball player and trainer.

Reuben Thomas is a freelance singer and computer scientist living in London. He took a BA in Mathematics with Computer Science from Cambridge University, as well as a doctorate in virtual machines. These days his computing interests center on contributions to a multitude of open source projects, with particular emphasis on improving the quality of mature software, and on automatic document processing. He is mostly employed as a classical baritone.

Robert Oates is a professional game programmer specializing in gameplay systems, artificial intelligence, and machine learning.

Roberto de Beauclair Seixas works with Research and Development at the Institute of Pure and Applied Mathematics (IMPA) in Rio de Janeiro, as member of the Vision and Computer Graphics Laboratory (Visgraf). He got his PhD in Computer Science at PUC-Rio, where he works with the Computer Graphics Technology Group (Tecgraf). From 1982 to 1998, he worked in the Computer Science Department at the National Laboratory for Scientific Computation (LNCC). His research interests include Scientific Visualization, Volume Rendering, Computer Graphics, High Performance Computing, Geometric Modeling, Military Warfare Simulations, GIS, and Medical Images.

Roberto Ierusalimschy is an Associate Professor at the Catholic University in Rio de Janeiro. He is the leading architect of Lua and the author of the book "Programming in Lua".

Sérgio Alvares Maffra is a MSc and Computer Engineer from PUC-Rio. He's been working with Lua at Tecgraf as a software developer for over a decade now.

Steve Gargolinski spent his early programming days hacking together small games built with code snippets from a QuickBasic programming manual. He has since evolved into a professional game developer, working as a member of the technical teams that produced the Zoo Tycoon 2 series, Star Trek: Legacy, and the upcoming Empire Earth III. Steve is currently working for Blue Fang Games as an AI Programmer. His interests include baseball, abstract strategy, practical AI, and walking in the woods.

Tobias Sülzenbrück and Christoph Beckmann are bachelor students of media systems at the Bauhaus-University of Weimar. Tobias fields of interests range from web development up to graphics programming. He has implemented a multi-agent system for simulating construction processes in Lua. Christoph is also interested in web development and is active in the research field of computer-supported cooperative work.

Tomás Guisasola works with Lua since 1995 when he developed with Roberto Ierusalimschy (his MSc advisor) the first implementation of the hooks mechanism and the debug facilities. Since then he worked mainly with CGILua as the platform for some administrative systems at PUC-Rio and also contributed

with the Kepler team in the development of LuaLDAP, LuaXMLRPC, LuaSOAP, LuaDoc, and LuaSQL.

Vadim Groznov began programming at the age of fourteen, was involved in database programming for a long time, and took part in the creation of a custom scripting language. He professionally develops video games since 2002. His extensive experience of system programming allowed for the design and realisation of complex architectural solutions for game tools at IceHill llc.

Wim Couwenberg holds a PhD in mathematics and is employed at the R&D department of the European printing and document company Océ, based in The Netherlands, where he organised the international Lua Workshop 2006. He has been using Lua in projects ranging from simple data processing scripts to entire networked applications.

Yuri Takhteyev is a doctoral student at the UC Berkeley School of Information studying the role of space in software development communities.

Han Zhao is a shareware programmer in Beijing, P.R. China. Before that he worked for a mobile-phone design house. Now he uses Lua and C++ for everyday programming: an isometric role-playing game engine, an action game, and a shareware product. He also maintains a bit-operation lib LuaBit on LuaForge.