



30 ANOS



PUC
RIO

1987

Tecgraf: parceria PUC-Rio + Petrobras



PUC
RIO

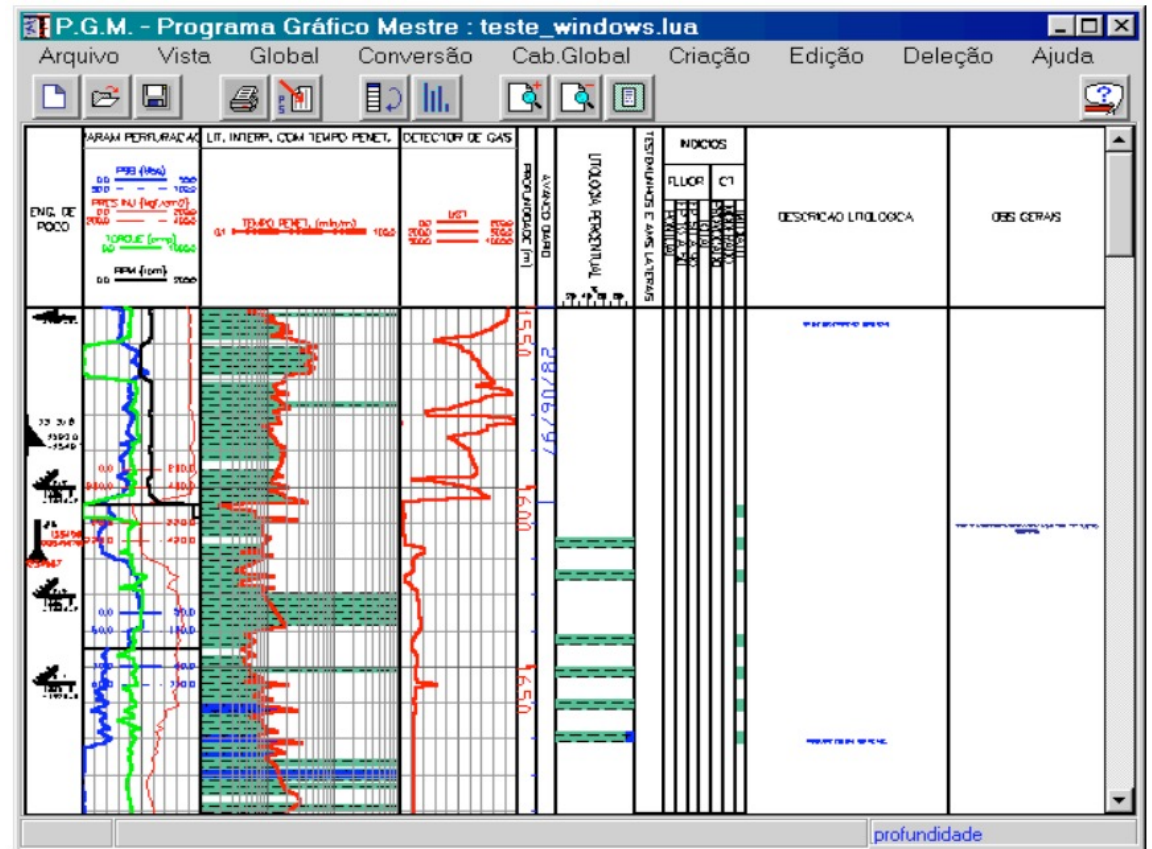
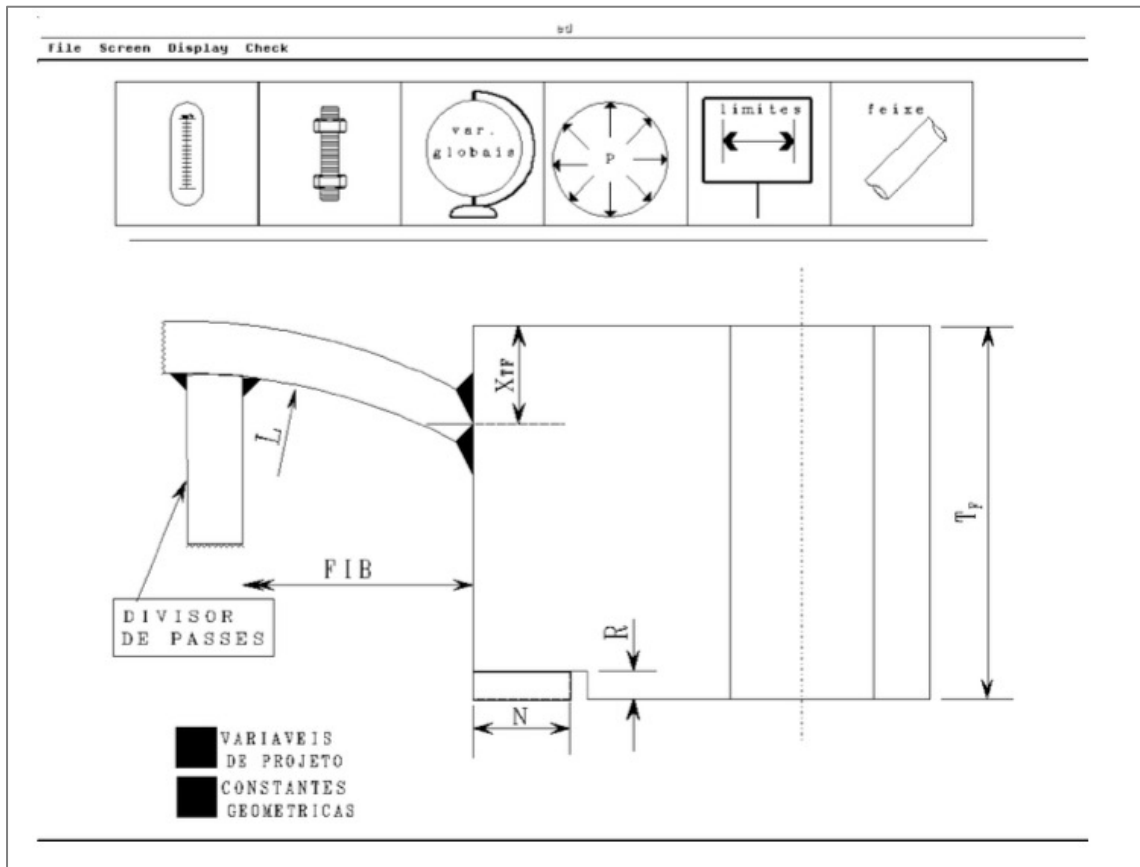


1992

origem: duas linguagens ad hoc

construção ágil de interfaces gráficas interativas

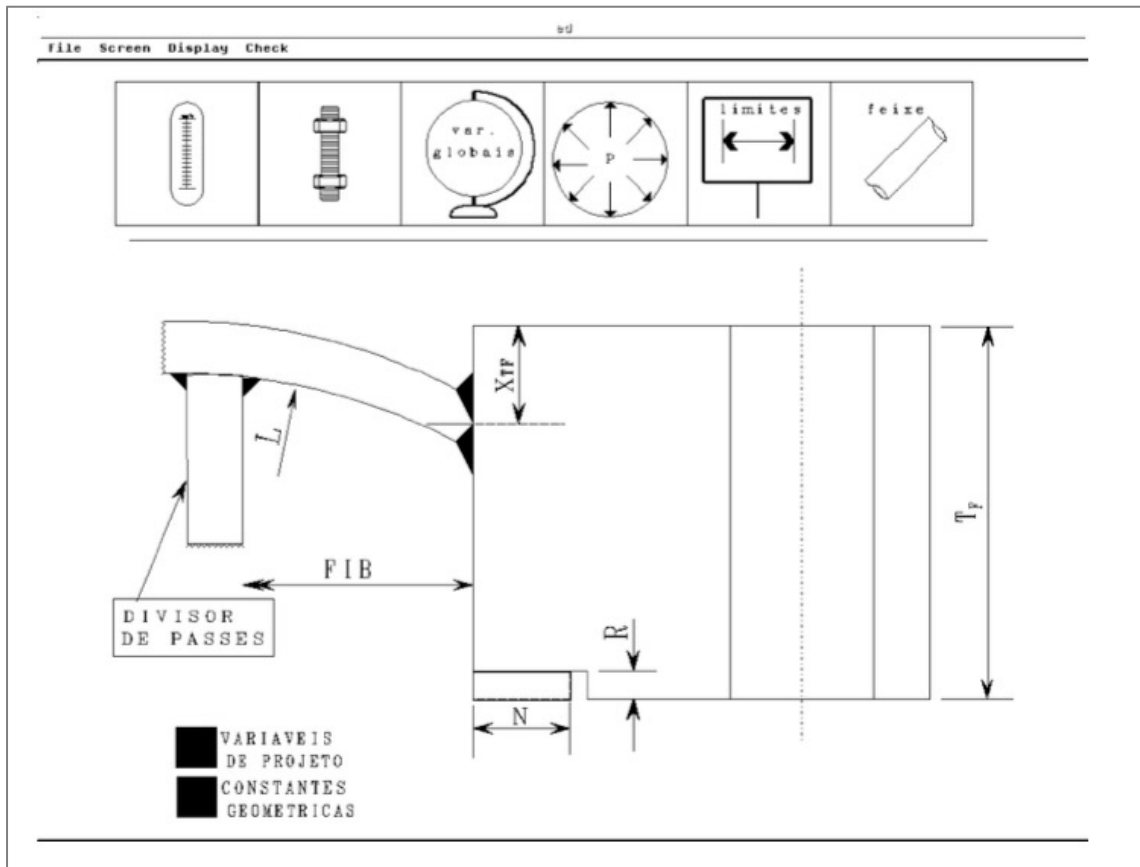
visualização customizada de dados de perfis geológicos



1992

DEL: Data Entry Language

construção ágil de
interfaces gráficas interativas



```
:e gasket "gasket properties"  
mat      s                # material  
d        f                0      # distance  
y        f                0      # settlement stress  
t        i                1      # facing type
```

```
:p gasket.m>30  
gasket.m<3000  
gasket.y>335.8  
gasket.y<2576.8
```


1993

Lua 1.0



1993

Lua 1.0

```
function check (object, class)
  local v = next(object,nil);
  while v ~= nil do
    if class[v] = nil then
      print("unknown field: " .. v)
    elseif type(object[v]) ~= class[v].type then
      print("wrong type for field " .. v)
    end
    v = next(object,v);
  end
  v = next(class,nil);
  while v ~= nil do
    if object[v] = nil then
      if class[v].default ~= nil then
        object[v] = class[v].default
      else print("field "..v.." not initialized")
      end
    end
    v = next(class,v);
  end
end
```

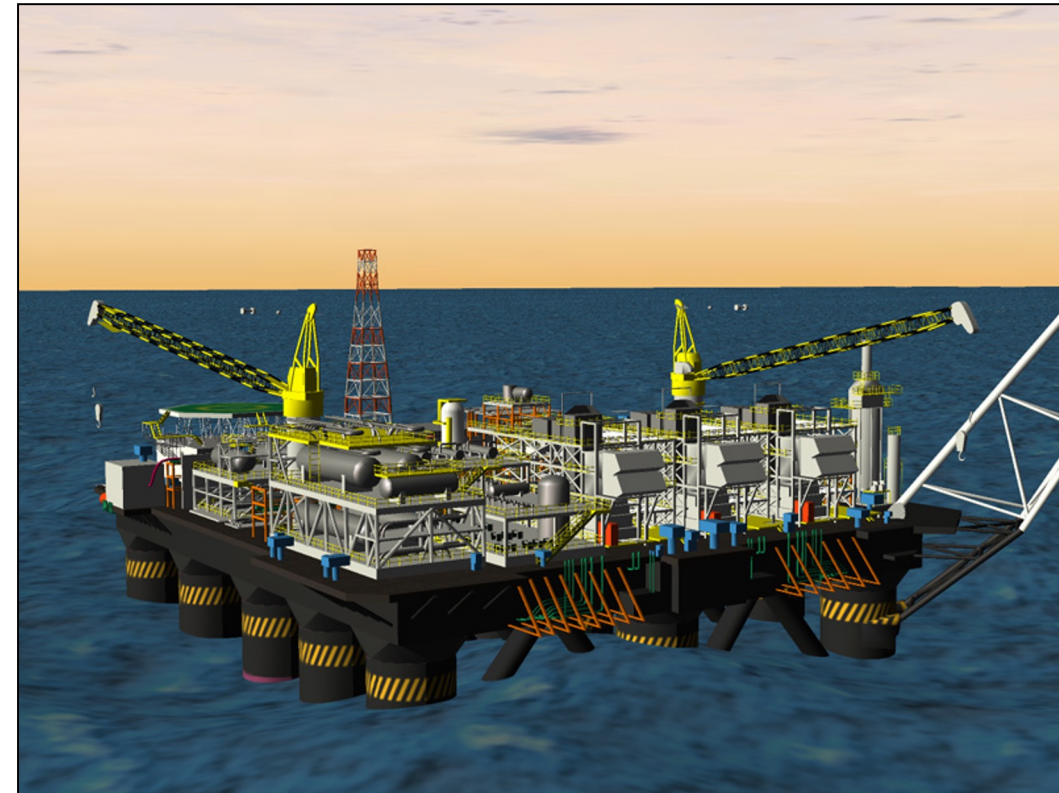
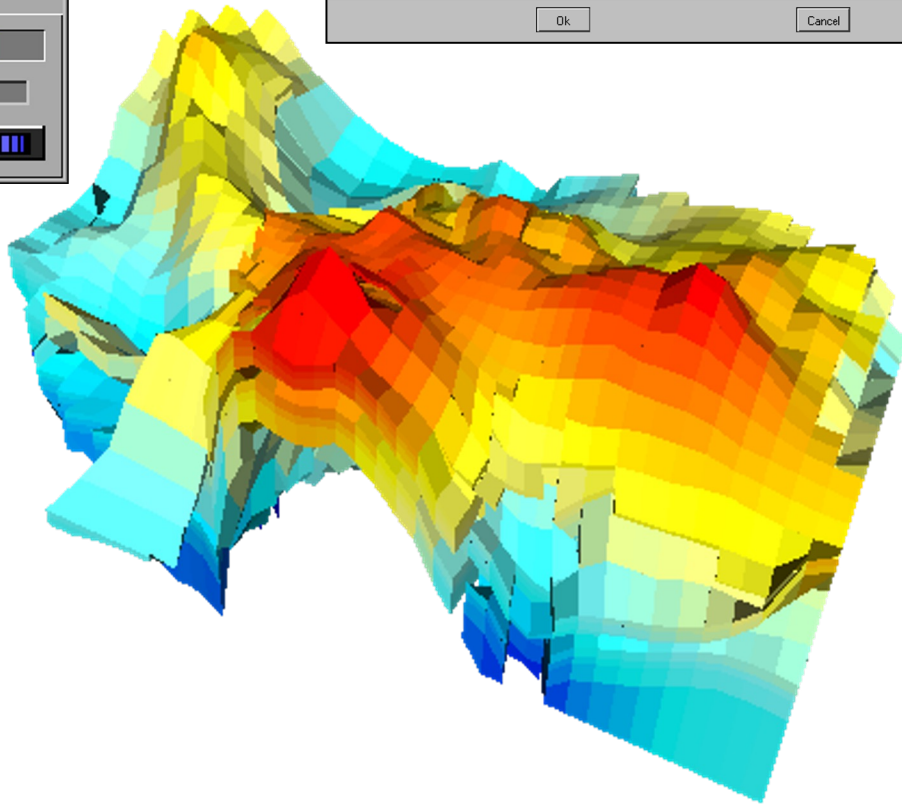
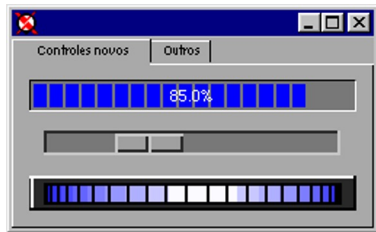
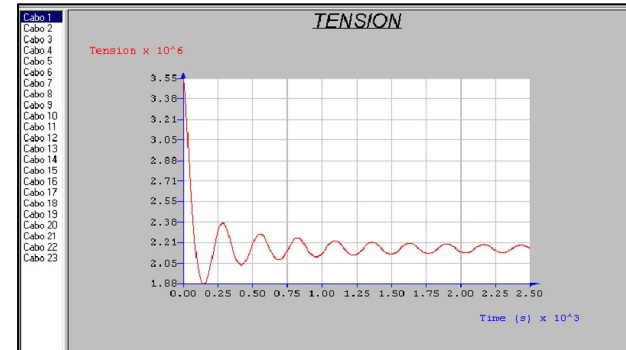
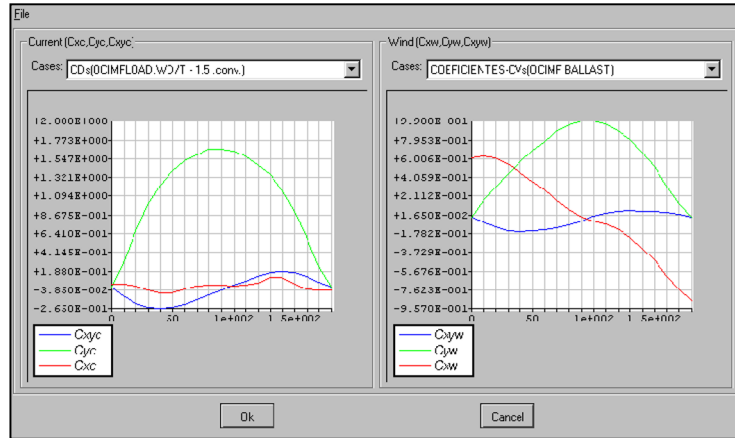
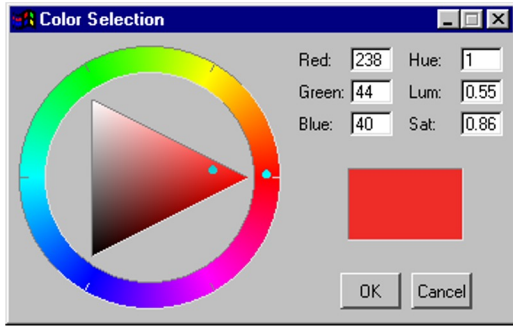
```
typetrilha = @{x = @{default = 0, type = "number"},
              y = @{default = 0, type = "number"},
              name = @{type = "string"}
             }

function trilha (t) check(t,typetrilha) end
t1 = @trilha{ x = 4, name = "1"}
t2 = @trilha{ z = 4, name = "2"}
t3 = @trilha{ x = 4, name = 3}
t4 = @trilha{}
```

```
unknown field: z
wrong type for field name
field name not initialized
```


1993

sucesso no Tecgraf



1994

primeiro artigo sobre Lua – SEMISH

The design and implementation of a language for extending applications

LUIZ HENRIQUE DE FIGUEIREDO, ROBERTO IERUSALIMSCHY, WALDEMAR CELES FILHO

TeCGraf–Grupo de Tecnologia em Computação Gráfica, ITS, PUC-Rio

Prédio do ITS, Rua Marquês de São Vicente 225, 22453-900 Rio de Janeiro, RJ, Brasil

{lhf,roberto,celes}@icad.puc-rio.br

Abstract. We describe the design and the implementation of Lua, a simple, yet powerful, language for extending applications. Although Lua is a procedural language, it has data description facilities, and has been extensively used in production for several tasks including user configuration, general-purpose data-entry, description of user interfaces, description of application objects, and storage of structured graphical metafiles.

Resumo. Descrevemos o projeto e a implementação de Lua, uma linguagem simples, porém poderosa, para extensão de aplicações. Embora procedural, Lua contém mecanismos para descrição de dados e tem sido largamente utilizada em produção para configuração pelo usuário, entrada de dados, descrição de interfaces, descrição de objetos da aplicação e armazenagem de *metafiles* gráficos estruturados.

1994

Lua 1.1

```
Newsgroups: comp.compilers,comp.lang.misc,comp.programming,comp.lang.c
From: lhf@csg.uwaterloo.ca (Luiz H de Figueiredo)
Subject: Announcing Lua, a language for extending applications
Date: Fri, 8 Jul 1994 11:51:45 GMT
```

This is the first public release of Lua.

* What is Lua?

Lua is a simple, yet powerful, language for extending applications.
Lua has been developed by TeCGraf, the Computer Graphics Technology Group
of PUC-Rio, the Catholic University of Rio de Janeiro, Brazil.
Dozens of industrial products developed by TeCGraf use Lua.

* Some features of Lua

Lua has a simple, pascal-like, syntax.
Variables need no declaration.
Lua has powerful data description constructs.
Functions can receive a variable number of arguments and can return multiple
values.
Lua programs are compiled into bytecodes, which are then interpreted to
simulate a virtual machine.
Lua is written in ANSI C and is completely portable.

1995

Lua 2.1

```
function check (object, class)
  local v = next(object,nil);
  while v ~= nil do
    if class[v] == nil then
      print("unknown field: " .. v)
    elseif type(object[v]) ~= class[v].type then
      print("wrong type for field " .. v)
    end
    v = next(object,v);
  end
  v = next(class,nil);
  while v ~= nil do
    if object[v] == nil then
      if class[v].default ~= nil then
        object[v] = class[v].default
      else print("field "..v.." not initialized")
      end
    end
    v = next(class,v);
  end
end
```

```
typetrilha = {x = {default = 0, type = "number"},
              y = {default = 0, type = "number"},
              name = {type = "string"}
             }
```

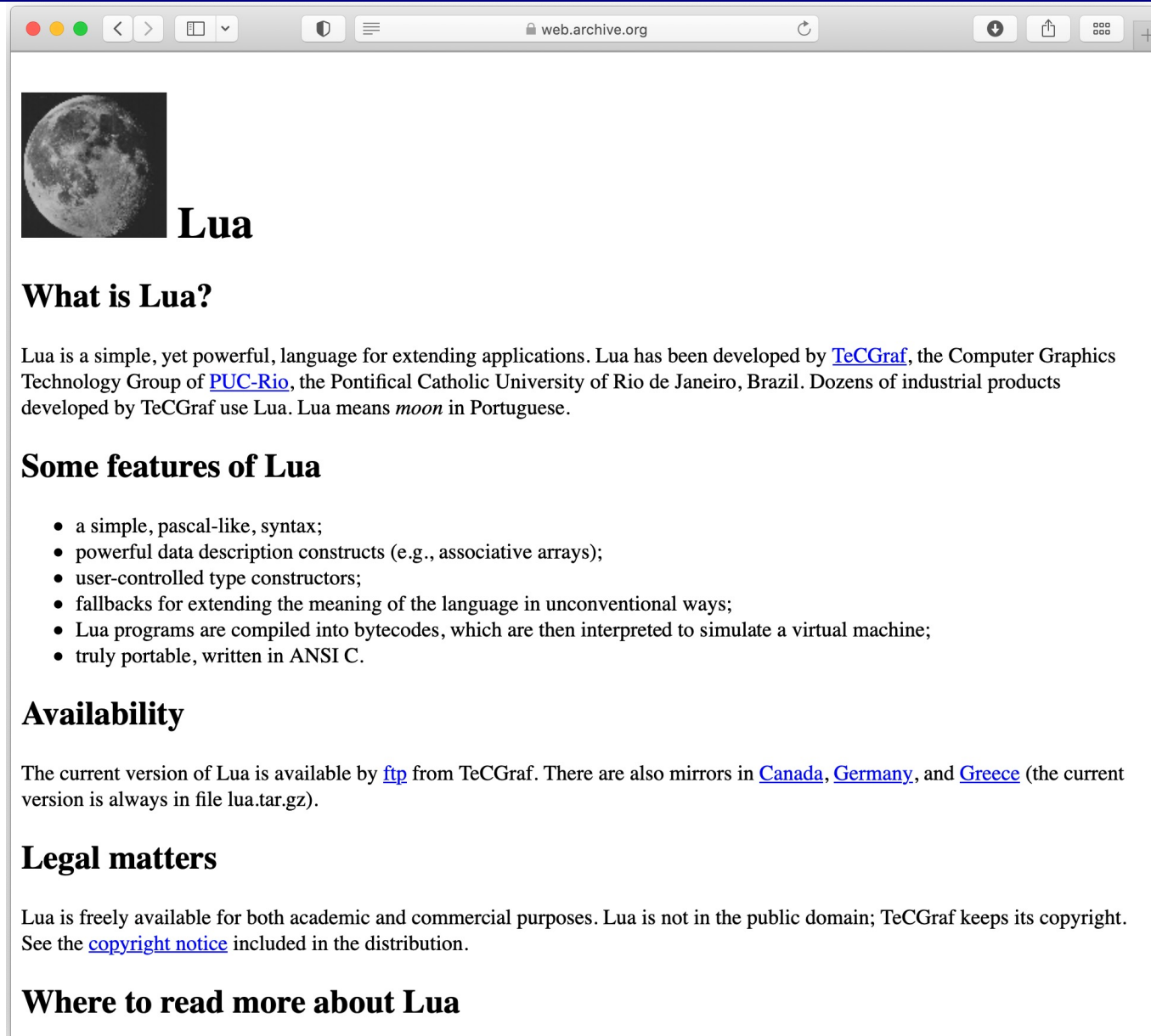
```
function trilha (t) check(t,typetrilha) end
t1 = trilha{ x = 4, name = "1"}
t2 = trilha{ z = 4, name = "2"}
t3 = trilha{ x = 4, name = 3}
t4 = trilha{}
```

```
unknown field: z
wrong type for field name
field name not initialized
```

1995

página web

www.inf.puc-rio.br/~roberto/luas.html



The screenshot shows a web browser window with the address bar displaying "web.archive.org". The page content includes a header image of the moon, followed by the title "Lua". Below the title, there are sections for "What is Lua?", "Some features of Lua", "Availability", "Legal matters", and "Where to read more about Lua".

Lua

What is Lua?

Lua is a simple, yet powerful, language for extending applications. Lua has been developed by [TeCGraf](#), the Computer Graphics Technology Group of [PUC-Rio](#), the Pontifical Catholic University of Rio de Janeiro, Brazil. Dozens of industrial products developed by TeCGraf use Lua. Lua means *moon* in Portuguese.

Some features of Lua

- a simple, pascal-like, syntax;
- powerful data description constructs (e.g., associative arrays);
- user-controlled type constructors;
- fallbacks for extending the meaning of the language in unconventional ways;
- Lua programs are compiled into bytecodes, which are then interpreted to simulate a virtual machine;
- truly portable, written in ANSI C.

Availability

The current version of Lua is available by [ftp](#) from TeCGraf. There are also mirrors in [Canada](#), [Germany](#), and [Greece](#) (the current version is always in file lua.tar.gz).

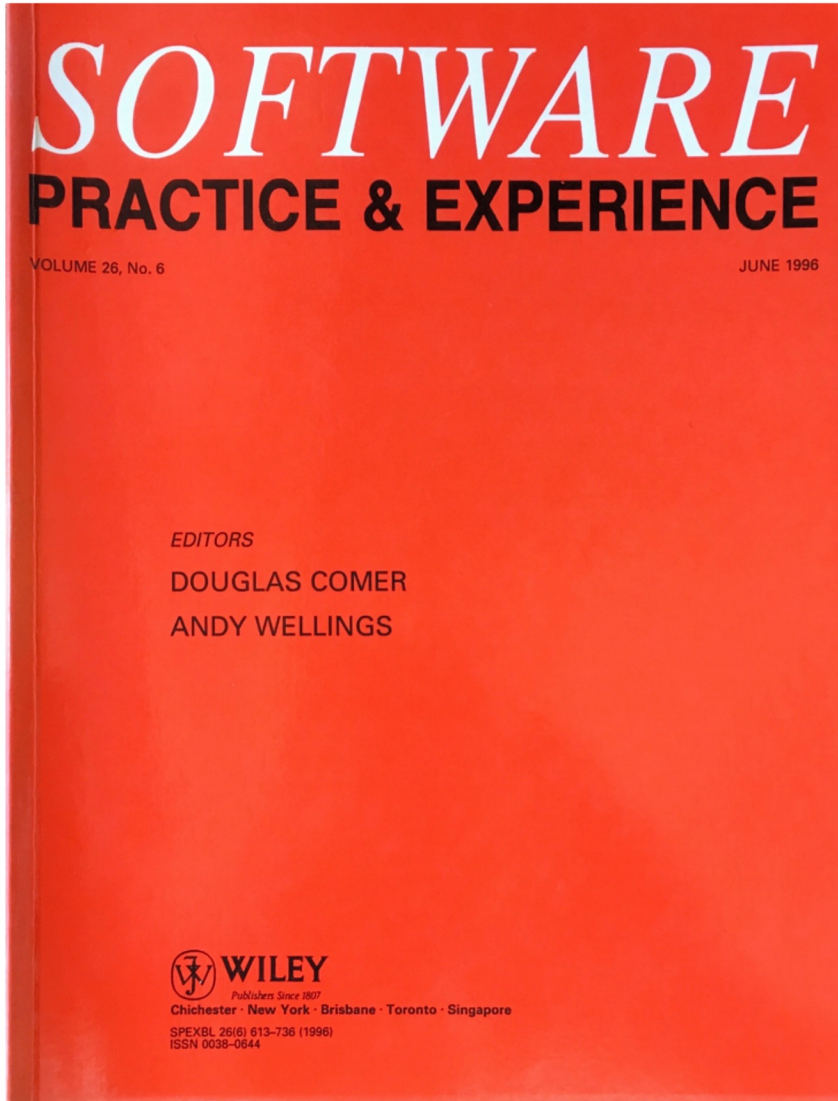
Legal matters

Lua is freely available for both academic and commercial purposes. Lua is not in the public domain; TeCGraf keeps its copyright. See the [copyright notice](#) included in the distribution.

Where to read more about Lua

1996

primeiro artigo em periódico



SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 26(6), 635–652 (JUNE 1996)

Lua—An Extensible Extension Language

ROBERTO IERUSALIMSCHY, LUIZ HENRIQUE DE FIGUEIREDO AND WALDEMAR CELES FILHO
TeCGraf, Computer Science Department, PUC–Rio, Rua M.S. Vicente 225, Rio de Janeiro, Brazil
(*email: {roberto,lhf,celes}@icad.puc-rio.br*)

SUMMARY

This paper describes Lua, a language for extending applications. Lua combines procedural features with powerful data description facilities, by using a simple, yet powerful, mechanism of *tables*. This mechanism implements the concepts of records, arrays and recursive data types (pointers), and adds some object-oriented facilities, such as methods with dynamic dispatching.

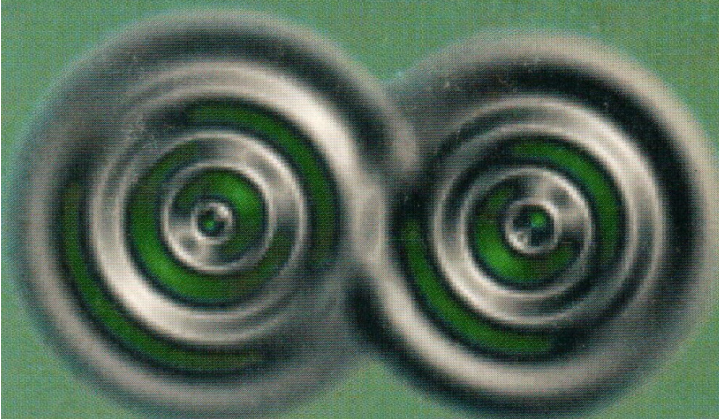
Lua presents a mechanism of *fallbacks* that allows programmers to extend the semantics of the language in some unconventional ways. As a noteworthy example, fallbacks allow the user to add different kinds of inheritance to the language.

Currently, Lua is being extensively used in production for several tasks, including user configuration, general-purpose data-entry, description of user interfaces, storage of structured graphical metafiles, and generic attribute configuration for finite element meshes.

1997

Prêmio Compaq

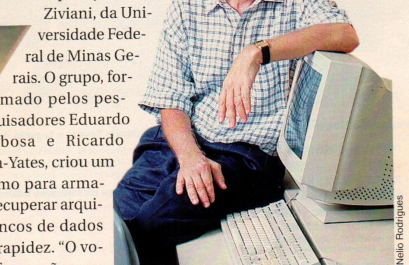
II PRÊMIO COMPAQ
DE ESTÍMULO À
PESQUISA E
DESENVOLVIMENTO
EM INFORMÁTICA



ANTENA DIGITAL

Cabeças premiadas nas universidades

Nos laboratórios de computação das universidades brasileiras dois temas têm atraído a atenção dos pesquisadores: a Internet e a forma de armazenar dados. Isso é o que demonstram os trabalhos vencedores da segunda edição do Prêmio Compaq de Estímulo à Pesquisa e Desenvolvimento em Informática, que tem o apoio de INFORMÁTICA EXAME. O primeiro lugar na categoria de pesquisa científica ficou com um grupo de estudantes orientados pelo professor de Computação Nívio



Nélio Rodrigues

Ziviani, da UFMG: algoritmo para facilitar a busca de dados

Ziviani, da Universidade Federal de Minas Gerais. O grupo, formado pelos pesquisadores Eduardo Barbosa e Ricardo Baeza-Yates, criou um algoritmo para armazenar e recuperar arquivos em bancos de dados com maior rapidez. "O volume de informações aumentou depois da disseminação da Internet, e criamos um software para guardar dados em memórias de acesso mais

veloz", explica Ziviani, professor titular do Departamento de Ciência da Computação da UFMG. Outras informações sobre o projeto estão no seguinte endereço: www.dcc.ufmg.br/~nivio. Na categoria de tecnologia aplicada, os vencedores foram os criadores de uma linguagem de programação chamada Lua. A linguagem foi desenvolvida em 1994 no laboratório TeCGraf, da PUC do Rio de Janeiro, onde a faculdade carioca entra em contato com o mundo real das necessidades das empresas. Os premiados foram Roberto Ierusalimsky, Luiz Henrique de Figueiredo e Waldemar Celes Filho. Lua é uma linguagem que não exige dos programadores um longo trabalho de pesquisa antes de iniciar a produção de um software. "Criamos uma forma de programar sem fazer estudos prévios. Basta usar o software e ir experimentando até chegar ao protótipo", diz Ierusalimsky, professor associado do Departamento de Informática da PUC carioca. Conheça melhor a Lua no site www.inf.puc-rio.br/~roberto/luas.html. Cada grupo dividirá entre si 20 000 reais e um total idêntico em equipamentos de informática.



Ella Parronina

Celes (à esq.), Ierusalimsky (sentado) e Figueiredo, da PUC-Rio: descobrindo a Lua

1996

Dr. Dobb's Journal

Java • C++ • Visual Basic

Dr. Dobb's
JOURNAL


#254 DECEMBER 1996

SOFTWARE
TOOLS FOR THE
PROFESSIONAL
PROGRAMMER

PORTABILITY & CROSS-PLATFORM DEVELOPMENT

- **LUA: AN EXTENSIBLE, PORTABLE EMBEDDED LANGUAGE**
- **DESIGNING A CROSS-PLATFORM WINHELP FUNCTION**
- **PORTING FROM DCE TO NT**

VISUAL DATABASE DEVELOPMENT
MULTITHREADING AND VISUAL BASIC
IMPLEMENTING JAVA DRAG-AND-DROP
INSIDE MFC'S UNDOCUMENTED
CSPLITTERWND CLASS
VIRTUAL AUDIO



\$4.95 (\$5.95 CANADA)

0 70992 35566 2 1.2

A Miller Freeman Publication

1997

LucasArts



From: Bret Mogilefsky <mogul@lucasarts.com>
To: "'lua@icad.puc-rio.br'" <lua@icad.puc-rio.br>
Subject: LUA rocks! Question, too.
Date: Thu, 9 Jan 1997 13:21:41 -0800

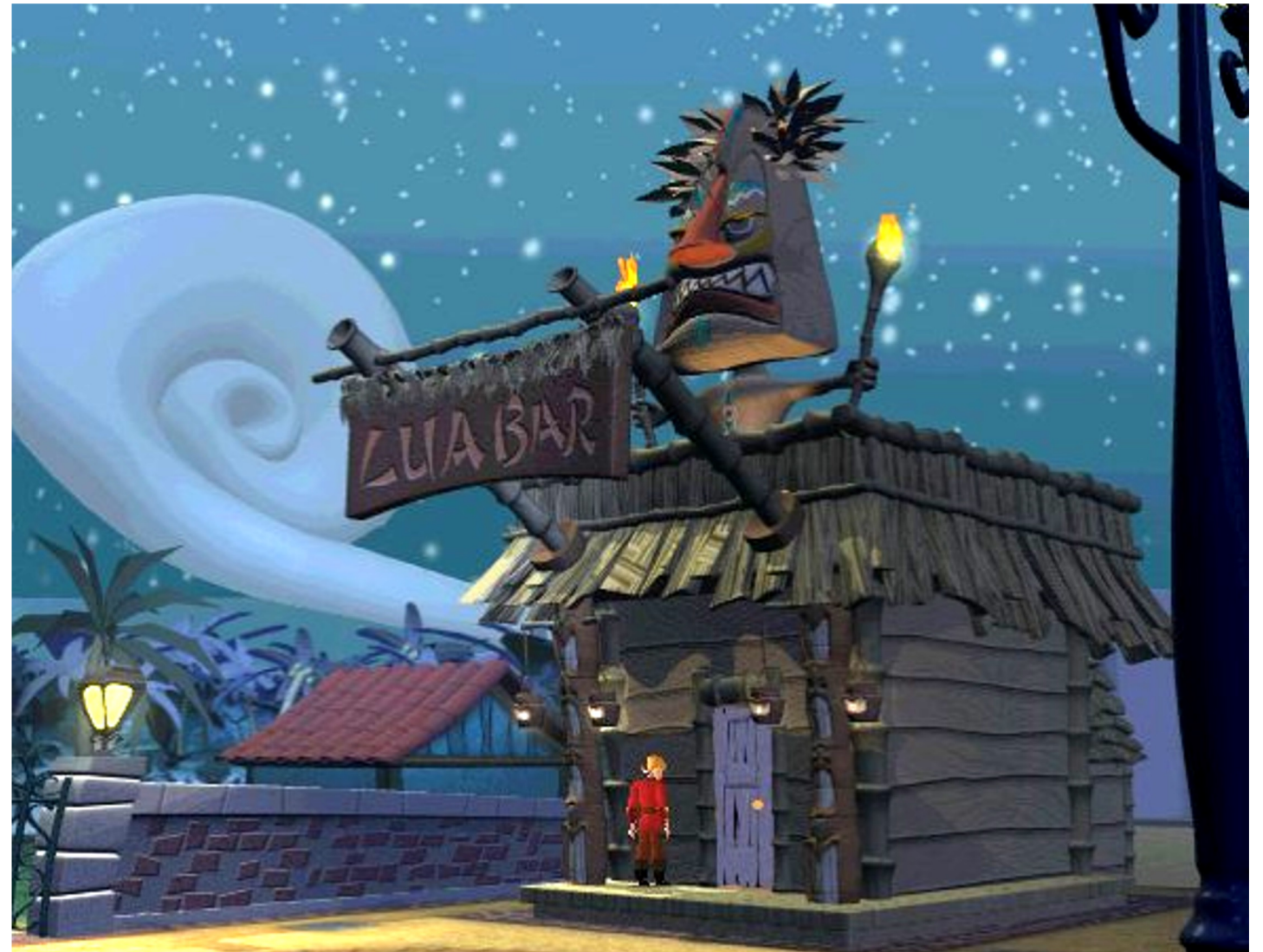
Hi there...

After reading the Dr. Dobbs article on Lua I was very eager to check it out, and so far it has exceeded my expectations in every way! It's elegance and simplicity astound me. Congratulations on developing such a well-thought out language.

Some background: I am working on an adventure game for the LucasArts Entertainment Co., and I want to try replacing our older adventure game scripting language, SCUMM, with Lua.

1998

LucasArts



2010

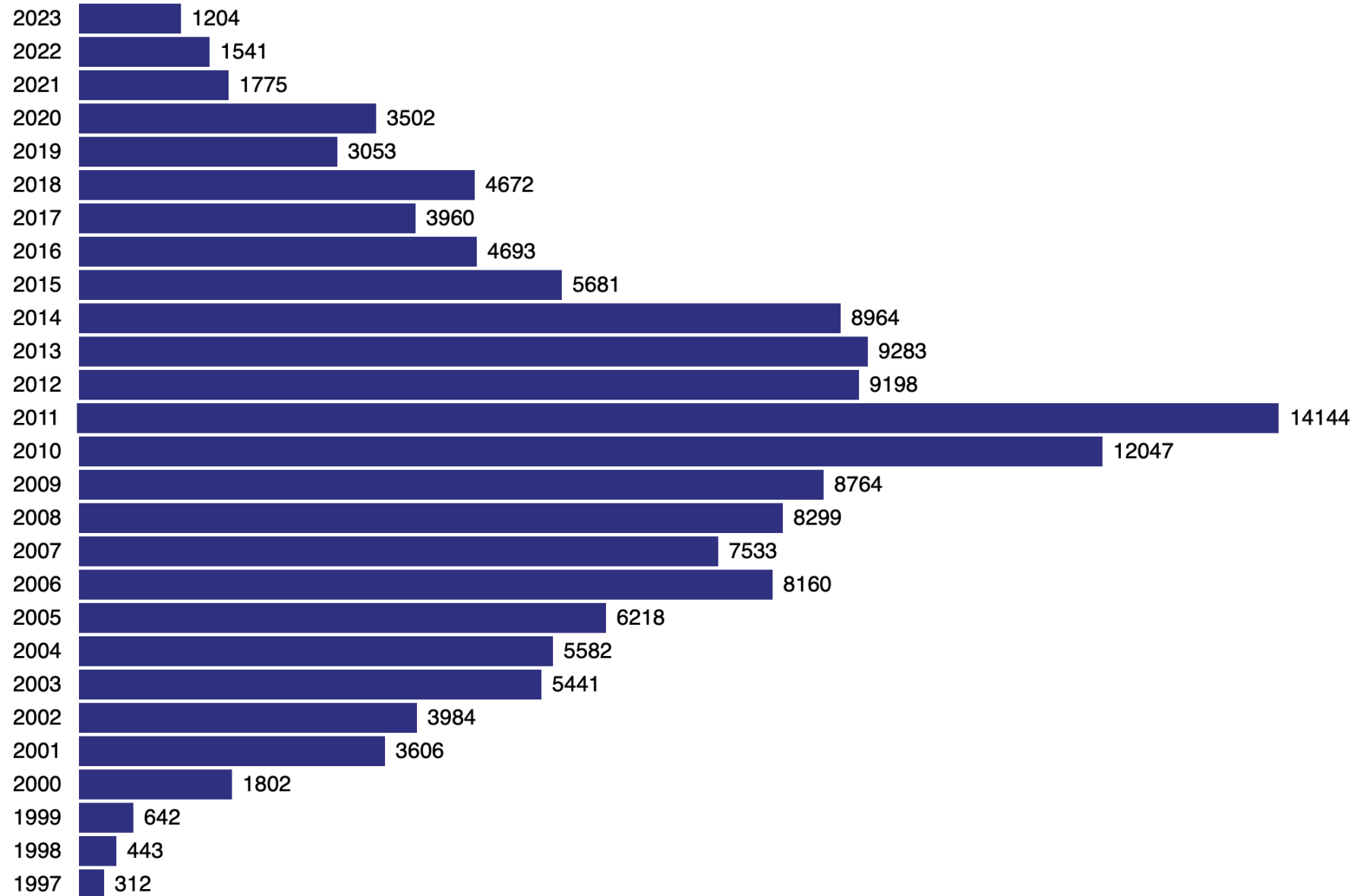
Diehard GameFAN: Hall of Fame Nomination



“Grim Fandango was the first game that shows Lua could not only be used to make a good game, but that it could be used to make some of the best games ever.”

1997

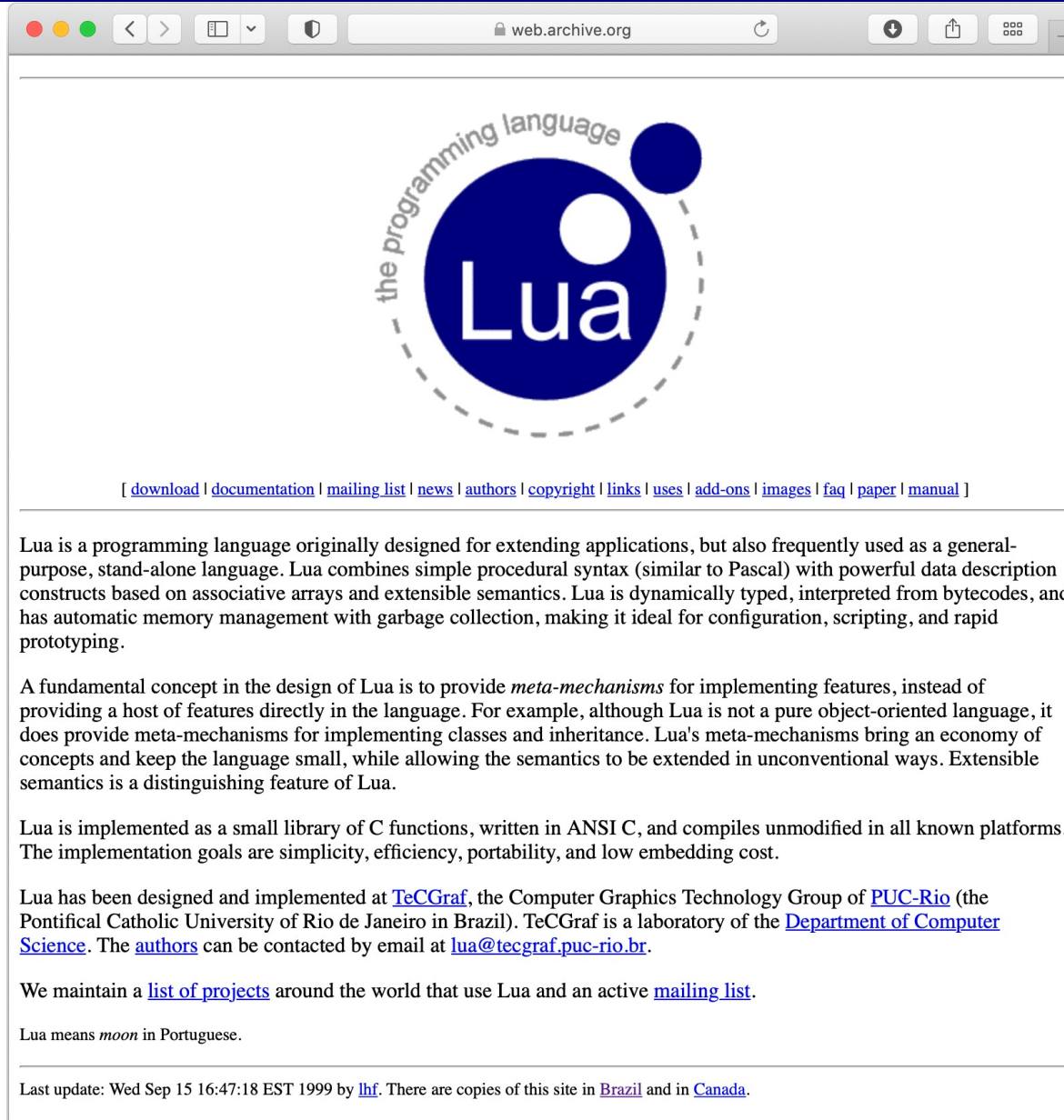
mailing list



1997

página web

www.tecgraf.puc-rio.br/lua



[[download](#) | [documentation](#) | [mailing list](#) | [news](#) | [authors](#) | [copyright](#) | [links](#) | [uses](#) | [add-ons](#) | [images](#) | [faq](#) | [paper](#) | [manual](#)]

Lua is a programming language originally designed for extending applications, but also frequently used as a general-purpose, stand-alone language. Lua combines simple procedural syntax (similar to Pascal) with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, interpreted from bytecodes, and has automatic memory management with garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

A fundamental concept in the design of Lua is to provide *meta-mechanisms* for implementing features, instead of providing a host of features directly in the language. For example, although Lua is not a pure object-oriented language, it does provide meta-mechanisms for implementing classes and inheritance. Lua's meta-mechanisms bring an economy of concepts and keep the language small, while allowing the semantics to be extended in unconventional ways. Extensible semantics is a distinguishing feature of Lua.

Lua is implemented as a small library of C functions, written in ANSI C, and compiles unmodified in all known platforms. The implementation goals are simplicity, efficiency, portability, and low embedding cost.

Lua has been designed and implemented at [TeCGraf](#), the Computer Graphics Technology Group of [PUC-Rio](#) (the Pontifical Catholic University of Rio de Janeiro in Brazil). TeCGraf is a laboratory of the [Department of Computer Science](#). The [authors](#) can be contacted by email at lua@tecgraf.puc-rio.br.

We maintain a [list of projects](#) around the world that use Lua and an active [mailing list](#).

Lua means *moon* in Portuguese.

Last update: Wed Sep 15 16:47:18 EST 1999 by [lhf](#). There are copies of this site in [Brazil](#) and in [Canada](#).

1998

novo logo



graphic design by Alexandre Nakonechnyj

2000

Lua na imprensa



Press clippings

Here are some articles about Lua that have appeared in the press. See also some [user quotes](#).

O Globo, 13 Sep 2022

Folha de São Paulo, 04 Mar 2022

O Globo, 10 Apr 2021

Folha de São Paulo, 21 Mar 2021

The Hosting Blog, 08 Jun 2017

Linux Magazine, 26 May 2016

O Globo, 07 Jul 2015

Canaltech, 09 Oct 2014

O Globo, 23 Jul 2013

Foreign Affairs, 21 Apr 2013

Wired, 19 Mar 2013

SFGATE, 30 Aug 2012

Game Developers Magazine, 13 Jan 2012

RTC Magazine, Dec 2011

O Globo, 26 Nov 2011

Veja, 31 Aug 2011

ITPRO, 24 May 2010

Computerworld, 11 Sep 2008

Dr. Dobb's, 8 Jan 2008

IEEE Software, Sep 2007

Revista Pesquisa FAPESP, Sep 2007

Linux Journal, 01 Jun 2007

Agência CT, 30 May 2007

Revista PROGRAMAR, May 2007

Universia, 24 May 2007

O Globo, 22 May 2007

SD Times, 15 Oct 2006

Jornal da PUC, 17 Aug 2006

ONLamp, 16 Feb 2006

Inovação Unicamp, 11 Aug 2005

Linux Magazine, Sep 2004

Linux Magazine, Aug 2004

O Globo, 19 Jul 2004

Unix Review, May 2004

Jornal do Brasil, 19 Apr 2004

Le Journal du Net, 06 Apr 2004

Jornal do Brasil, 08 Sep 2002

Unix Review, May 2002

Developers' Magazine, Mar 2001

Veja Rio, 19 Feb 2001

Exame, 13 Dec 2000

C Magazine, Dec 2000

Globo.com, 16 Nov 2000

Hipermidia, 11 and 16 Nov 2000

Jornal da PUC, Nov 2000

ISTOÉ Dinheiro, 25 Oct 2000

O Globo, 16 Oct 2000

O Globo, 02 Oct 2000

Jornal do Commercio, 01 Oct 2000

Archive, Oct 2000

Gazeta Mercantil, 29 Sep 2000

Adventurer Paradise, 20 Aug 2000

SunWorld, Aug 1998

Informática Exame, Nov 1997

'Nossa linguagem é totalmente livre e flexível'

Criador da linguagem Lua, usada como ferramenta por geólogos e até designers, explica seu funcionamento

ENTREVISTA
Roberto Ierusalimsky

só conferir com Roberto Ierusalimsky — um dos criadores da linguagem de programação Lua, surgi-

da no Tecgraf (Grupo de Tecnologia em Computação Gráfica, parceria da PUC-RJ com a Petrobras). Roberto, engenheiro de sistemas com pós-doutorado na Universidade de Waterloo, no Canadá, professor associado do departamento de informática da PUC-

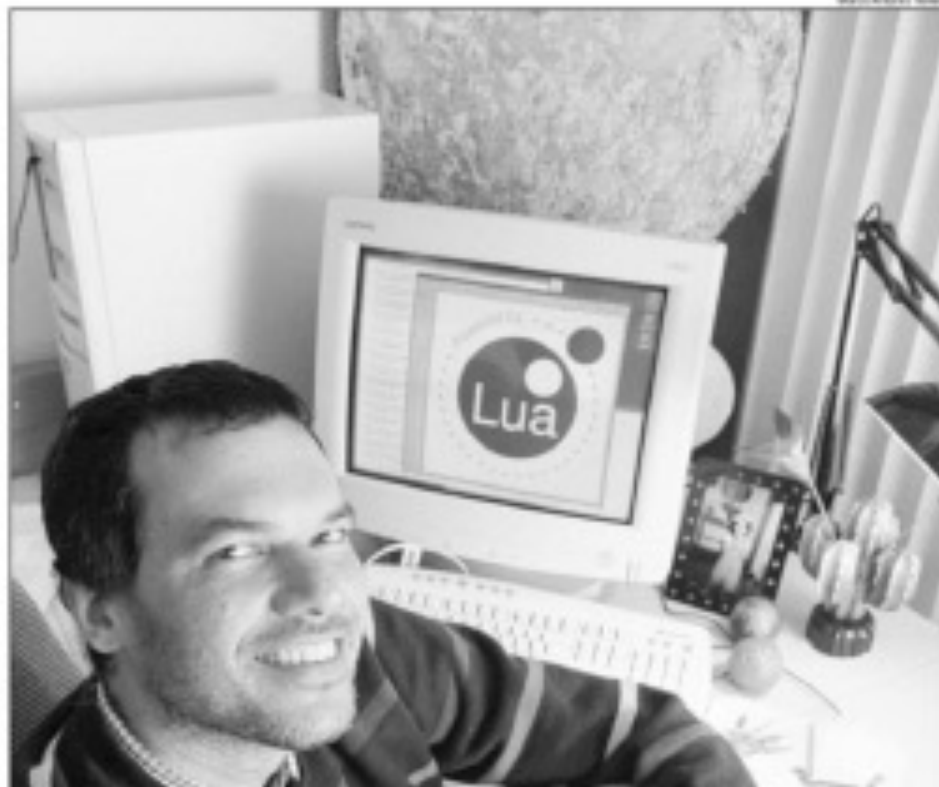
RJ e consultor do Tecgraf, criou a linguagem junto com os companheiros Waldemar Celes e Luiz Henrique de Figueiredo e a viu crescer mundo. Ele nos revelou a trajetória do software em seu escritório, sob os auspícios do logotipo "lunar" da linguagem.

RJ e consultor do Tecgraf, criou a linguagem junto com os companheiros Waldemar Celes e Luiz Henrique de Figueiredo e a viu crescer mundo. Ele nos revelou a trajetória do software em seu escritório, sob os auspícios do logotipo "lunar" da linguagem.

André Machado

O GLOBO: Por que o nome Lua? E algum acrônimo?
ROBERTO IERUSALIMSKY: Quando a criamos, em 1993, planejamos uma linguagem mais, chamada SOL (Simple Object Language), mas depois desistimos dela e pensamos em reduzi-la. Al, alguém sugeriu: já que vocês não fazem algo menor do que o Sol... E veio o nome Lua.

• Como ela funciona?
ROBERTO: A língua básica é ser uma linguagem que sacrifica um pouco eficiência de execução (isto é tão eficiente como, por exemplo, C e C++, mas pesada) em troca de flexibilidade. Lua é mais fácil de programar, você não tem que se preocupar com certas coisas. Mas sua maior diferencial é ter uma interface muito fácil com C e C++. Então, a língua central é uma máquina: some partes do seu programa em C e partes em Lua, de modo que, onde você precisa de mais eficiência da máquina, usa o C; e onde precisa de mais flexibilidade para experimentação, usa Lua. Ela permite um equilíbrio melhor entre essas duas coisas, além de ser mais fácil de modificar. Pode-se, por exemplo, meter na instalação sem ter que recompilá-la... E também é uma linguagem com



Criando uma ponte com a Web

• A linguagem Lua tem vários "derivados". O principal deles é o CGI/Lua, que trabalha com aplicações Internet. Um dos principais usuários de CGI/Lua no Brasil é a Fábrica Digital, que escreveu sua ferramenta de edição Web, o Publifácil, todo em Lua e usa a linguagem derivada para conectá-la à Grade Breda. Quem explica o funcionamento de CGI/Lua é André Carnevali, responsável pela parte de desenvolvimento de software na empresa.

— CGI/Lua é um programa escrito em C onde você escreve trechos escritos em Lua. É o executável do CGI/Lua faz toda a comunicação desse trecho Lua com o protocolo http da Internet. Ou seja, ele faz o papel do CGI. — diz.

É qual é a vantagem de usar CGI/Lua? Segundo André, é o controle que se ganha sobre o ciclo de

2004

Jornal do Brasil

Internet

internet@jb.com.br

JORNAL DO BRASIL ☆ SEGUNDA-FEIRA, 19 DE ABRIL DE 2004

Linguagem de programação criada na PUC-Rio é usada em ônibus espacial, processador, UTI e na criação do melhor da diversão eletrônica



LUIZ HENRIQUE, Roberto e Waldemar criaram e mantêm a linguagem em evolução, com o apoio da comunidade na internet



Brilho lunar

2001

página web

www.lua.org



The Evolution of an Extension Language: A History of Lua

Roberto Ierusalimschy

Luiz Henrique de Figueiredo

Waldemar Celes

TeCGraf, Department of Computer Science, PUC-Rio

`lua@tecgraf.puc-rio.br`

Abstract

Since its inception, in 1993, the Lua programming language has gone far beyond our most optimistic expectations. In this paper, we describe the trajectory of Lua, from its creation as an in-house language for two specific projects, until Lua 4.0, released in November 2000. We discuss the evolution of some of its concepts and the main landmarks in its implementation.

2002

nova licença – free software



License

Lua is free software distributed under the terms of the [MIT license](#) reproduced here. Lua may be used for any purpose, including commercial purposes, at absolutely no cost. No paperwork, no royalties, no GNU-like "copyleft" restrictions, either. Just [download](#) it and use it.

Lua is certified [Open Source](#) software. Its license is simple and liberal and is [compatible with GPL](#). Lua is not in the public domain and [PUC-Rio](#) keeps its copyright.



The spirit of the Lua license is that you are free to use Lua for any purpose at no cost without having to ask us. The only requirement is that if you do use Lua, then you should give us credit by including the copyright notice somewhere in your product or its documentation. A nice, but optional, way to give us further credit is to include a [Lua logo](#) and a [link to our site](#) in a web page for your product.

The Lua language is entirely designed, implemented, and maintained by a [team](#) at [PUC-Rio](#) in Brazil. The implementation is not derived from licensed software.

Before Lua 5.0, Lua used its own license, which was very close to the [zlib license](#) and others, but not quite the same. Check the [source distribution](#) for the exact license text for each version before Lua 5.0. Nevertheless, if you wish to use those old versions, you may hereby assume that they have all been re-licensed under the MIT license as described above.

last update: mon may 15 14:17:14 utc 2023

Copyright © 1994–2023 Lua.org, PUC-Rio.

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.

2004

LabLua



LabLua

programming language research

The Implementation of Lua 5.0

Roberto Ierusalimschy¹, Luiz Henrique de Figueiredo², Waldemar Celes¹

¹Department of Computer Science, PUC-Rio, Rio de Janeiro, Brazil.

²IMPA–Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil.

Abstract. We discuss the main novelties of the implementation of Lua 5.0: its register-based virtual machine, the new algorithm for optimizing tables used as arrays, the implementation of closures, and the addition of coroutines.

1. Introduction

Lua was born in an academic laboratory as a tool for in-house software development but somehow was adopted by several industrial projects around the world and is now widely used in the game industry.¹

How do we account for this widespread use of Lua? We believe that the answer lies in our design and implementation goals: to provide an embeddable scripting language that is simple, efficient, portable, and lightweight. These have been our main goals since the birth of Lua in 1993 and they have been respected during its evolution. (For a history of the development of Lua up to just before the release of Lua 5.0, see [12].) These features, plus the fact that Lua has been designed from the start to be embedded into larger applications, account for its early adoption by the industry.²

2005

workshop na sede da Adobe



Wed July 27

- 9:00 breakfast
- 9:30 **Io and Lua**
Steve Dekorte
- 10:00 **Lua and BREW** cancelled
Marc Nijdam
- 10:30 coffee break
- 11:00 **The Implementation of Lua**
Roberto Ierusalimsky
- 12:00 lunch
- 1:30 **Luasocket Behind the Scenes**
Diego Nehab
- 2:00 **It's all glue: Experiences building a desktop application with Lua**
Mark Hamburg
- 2:30 **Lua and Instant Messaging**
Matthew Burke
Embedding Lua into LabView
Albert-Jan Brouwer
Using Lua to Build Domain-Specific Languages
Thomas Wrench
Lua in ATE
Marius Gheorghe
- 3:30 coffee break
- 4:00 open discussion
- 5:00 closing

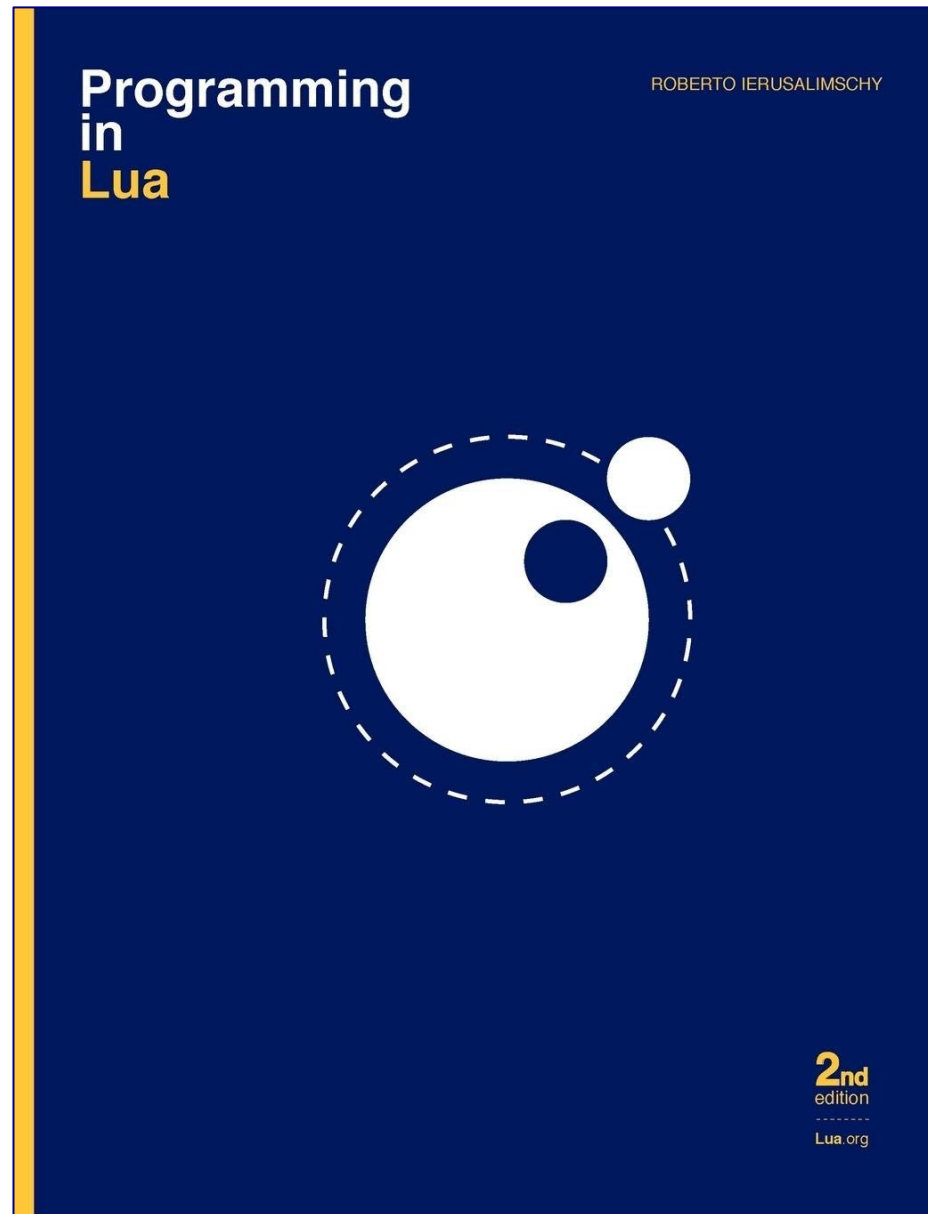
Thu July 28

- 9:00 breakfast
- 9:30 **Embedding Lua in Grim Fandango**
Bret Mogilefsky
- 10:00 **Psychonauts and Lua: a case study**
Paul Du Bois
- 10:30 coffee break
- 11:00 **The Novelities of Lua 5.1**
Roberto Ierusalimsky
- 12:00 lunch
- 1:30 **The Kepler Project**
André Carregal
- 2:00 **A Macro Facility for Lua**
Luiz Henrique de Figueiredo
- 2:30 **Lua for Molecular Biology**
Yutaka Ueno
LuaSynth
Alex Mohr
Lua Integrated into Embedded Linux
Glenn Edgar
A Lua Decompiler
Hisham Muhammad
- 3:30 coffee break
- 4:00 **The Future of Lua**
roundtable
- 5:00 closing



2006

Programming in Lua – 2ed



2007

ACM History of Programming Languages

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Roberto Ierusalimschy
Luiz Henrique de Figueiredo
Waldemar Celes

The Evolution of Lua

Roberto Ierusalimschy

Department of Computer Science,
PUC-Rio, Rio de Janeiro, Brazil

roberto@inf.puc-rio.br

Luiz Henrique de Figueiredo

IMPA–Instituto Nacional de
Matemática Pura e Aplicada, Brazil

lhf@impa.br

Waldemar Celes

Department of Computer Science,
PUC-Rio, Rio de Janeiro, Brazil

celes@inf.puc-rio.br



Abstract

We report on the birth and evolution of Lua and discuss how it moved from a simple configuration language to a versatile, widely used language that supports extensible semantics, anonymous functions, full lexical scoping, proper tail calls, and coroutines.

Categories and Subject Descriptors K.2 [HISTORY OF COMPUTING]: Software; D.3 [PROGRAMMING LANGUAGES]

1. Introduction

Lua is a scripting language born in 1993 at PUC-Rio, the Pontifical Catholic University of Rio de Janeiro in Brazil. Since then, Lua has evolved to become widely used in all kinds of industrial applications, such as robotics, literate

ing languages offer associative arrays, in no other language do associative arrays play such a central role. Lua tables provide simple and efficient implementations for modules, prototype-based objects, class-based objects, records, arrays, sets, bags, lists, and many other data structures [28].

In this paper, we report on the birth and evolution of Lua. We discuss how Lua moved from a simple configuration language to a powerful (but still simple) language that supports extensible semantics, anonymous functions, full lexical scoping, proper tail calls, and coroutines. In §2 we give an overview of the main concepts in Lua, which we use in the other sections to discuss how Lua has evolved. In §3 we relate the prehistory of Lua, that is, the setting that led to its creation. In §4 we relate how Lua was born, what its original design goals were, and what features its first version had. A discussion of how and why Lua has evolved is given in §5. A detailed discussion of the evolution of selected features

2008

Lua Programming Gems

Lua programming gems



edited by
Luiz Henrique de Figueiredo
Waldemar Celes
Roberto Ierusalimschy

Lua.org

“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*. [...] its individual chapters get across ideas that simply aren’t explained anywhere else.”
—Cameron Laird

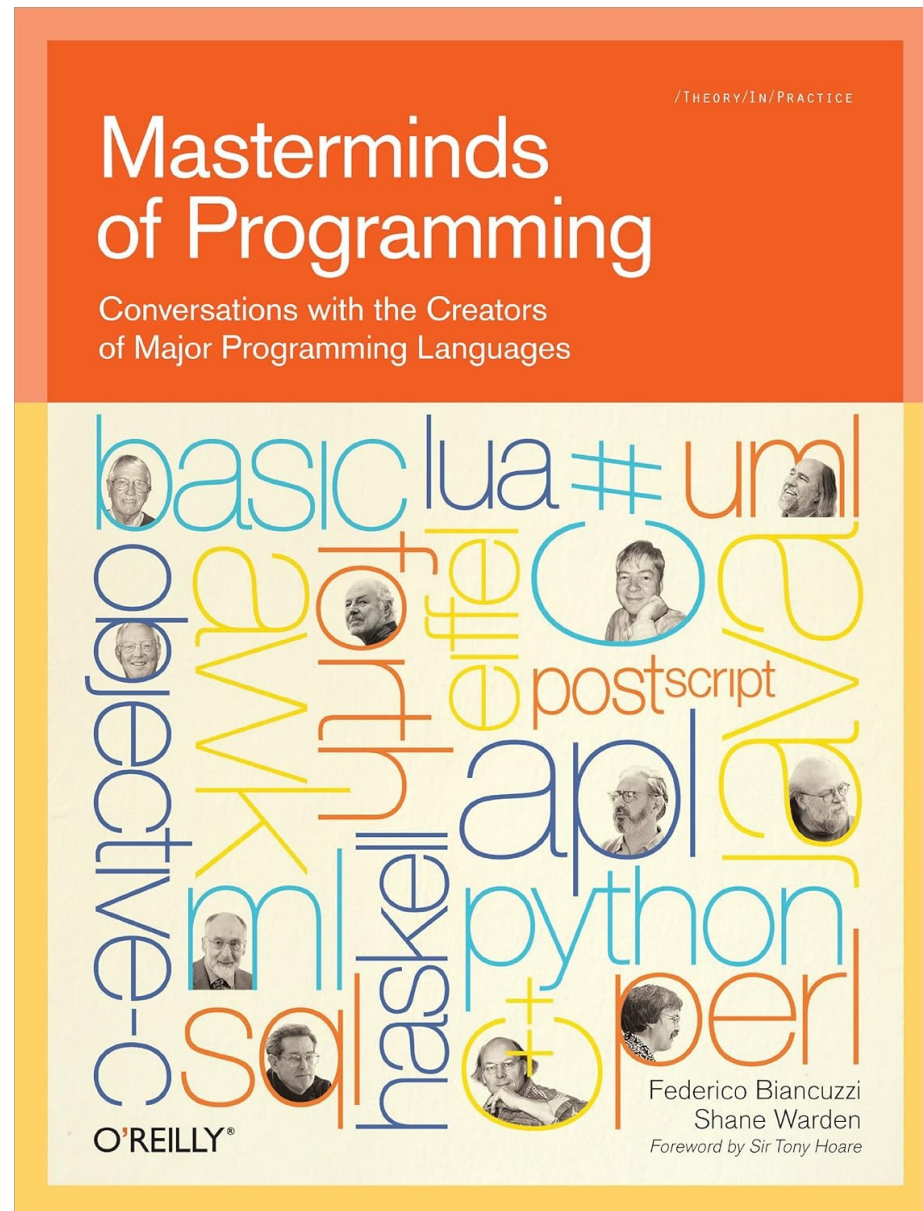
“If this book had been around during Lightroom’s development, we probably would have happily adopted some of the techniques it presents while simply taking inspiration from others. [...] The Lua community and particularly new Lua users can be happy to now have a field guide that maps out some of the trails. [...] 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.”
—Mark Hamburg

2009

Masterminds of Programming

CHAPTER SEVEN

Lua



Lua is a very small, self-contained dynamic language created by Roberto Ierusalimsky, Luiz Henrique de Figueiredo, and Waldemar Celes in 1993. Lua's small set of powerful features and easy-to-use C API make the language easy to embed and extend to express domain-specific concepts. Lua is prominent in the world of proprietary software, where games such as Blizzard's *World of Warcraft* and Crytek GmbH's *Crysis*, as well as Adobe's Photoshop Lightroom, use it for scripting and UI work. Its predecessors are Lisp, Scheme, and perhaps AWK; it has design similarities to JavaScript, Icon, and Tcl.

How the embeddability of Lua impacted its design.

BY ROBERTO IERUSALIMSKY, LUIZ HENRIQUE DE FIGUEIREDO,
AND WALDEMAR CELES

Passing a Language Through the Eye of a Needle

SCRIPTING LANGUAGES ARE an important element in the current landscape of programming languages. A key feature of a scripting language is its ability to integrate with a system language.⁷ This integration takes two main forms: *extending* and *embedding*. In the first form, you extend the scripting language with libraries and functions written in the system language and write your main program in the scripting language. In the second form, you embed the scripting language in a host program (written in the system language) so that the host can run scripts and call functions defined in the scripts; the main program is the host program. In this setting, the system language is usually called the host language.



```
object *next p->gch.next; /* save each node in the list  
newhash[h1] = p = next;(p)->hash = newhash[h1]; /* chain  
next */(unsigned int)gco2ts(p) * object *next p->gch.next;  
newhash[h1] = p = next;(p)->hash = newhash[h1]; /* chain  
save next */(unsigned int)gco2ts(p) * object *next p->gch.next;  
newhash[h1] = p = next;(p)->hash = newhash[h1]; /* chain  
newsize) == lmod(h, newsize); p->hash = newhash[h1]; /* chain  
newsize); p->hash = newhash[h1]; /* chain
```


2011

Front Line Award Game Developers Magazine





waze 



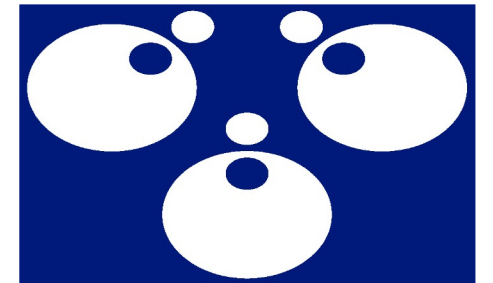
QSC



WIRESHARK



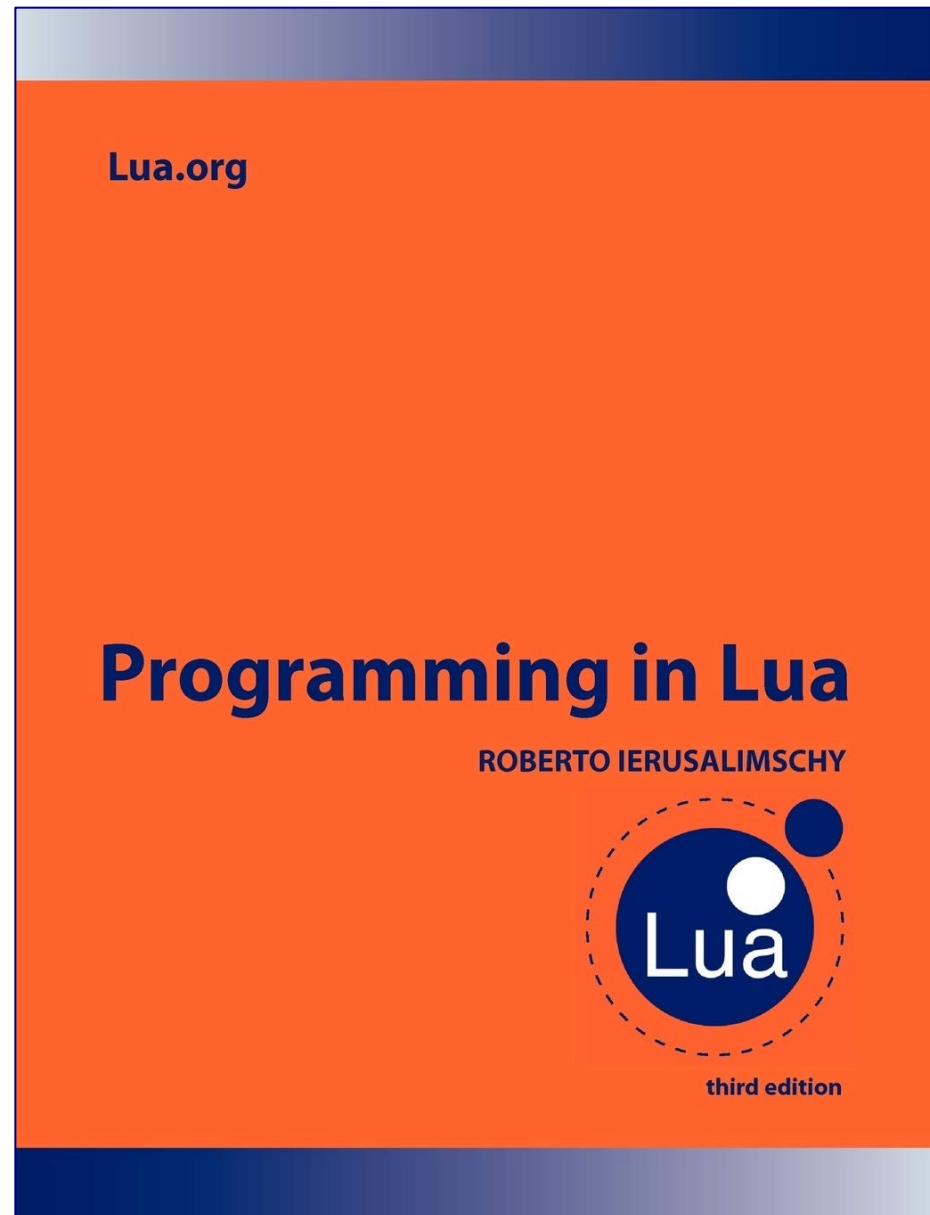
sas



neovim

2013

Programming in Lua – 3ed



2013

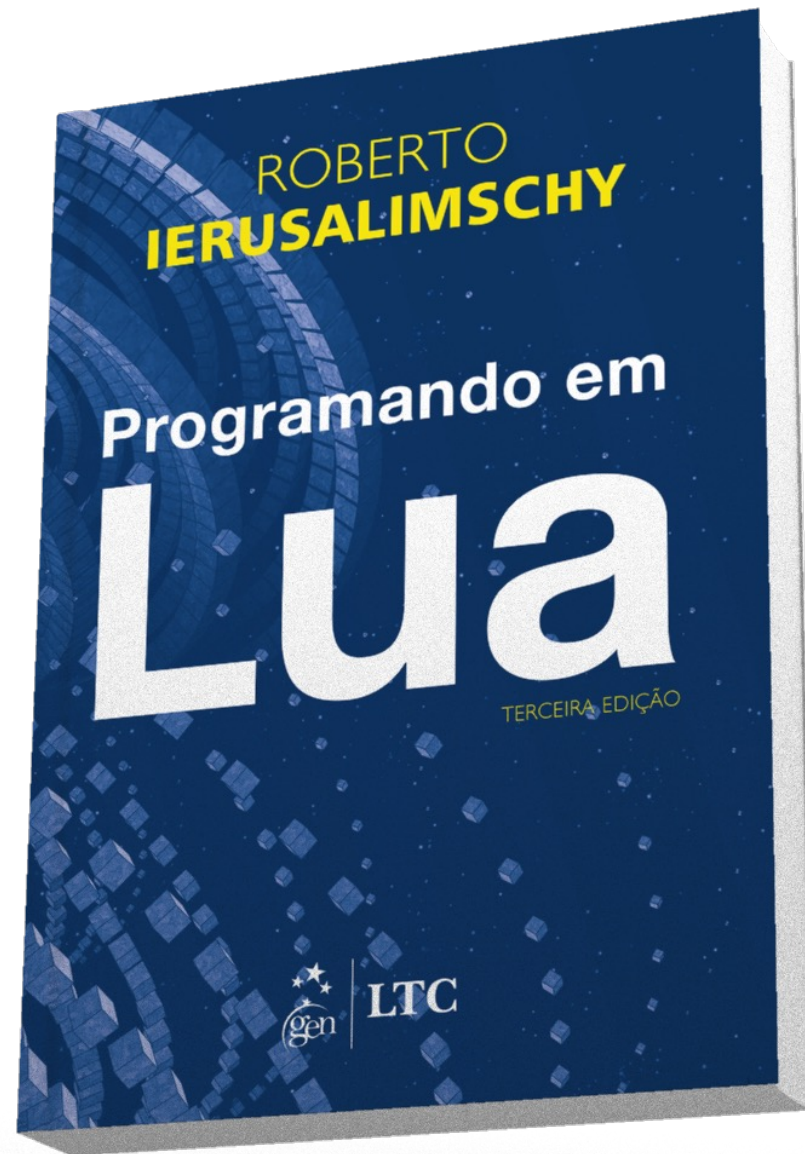
Prêmio Mérito Científico



Sociedade Brasileira
de Computação

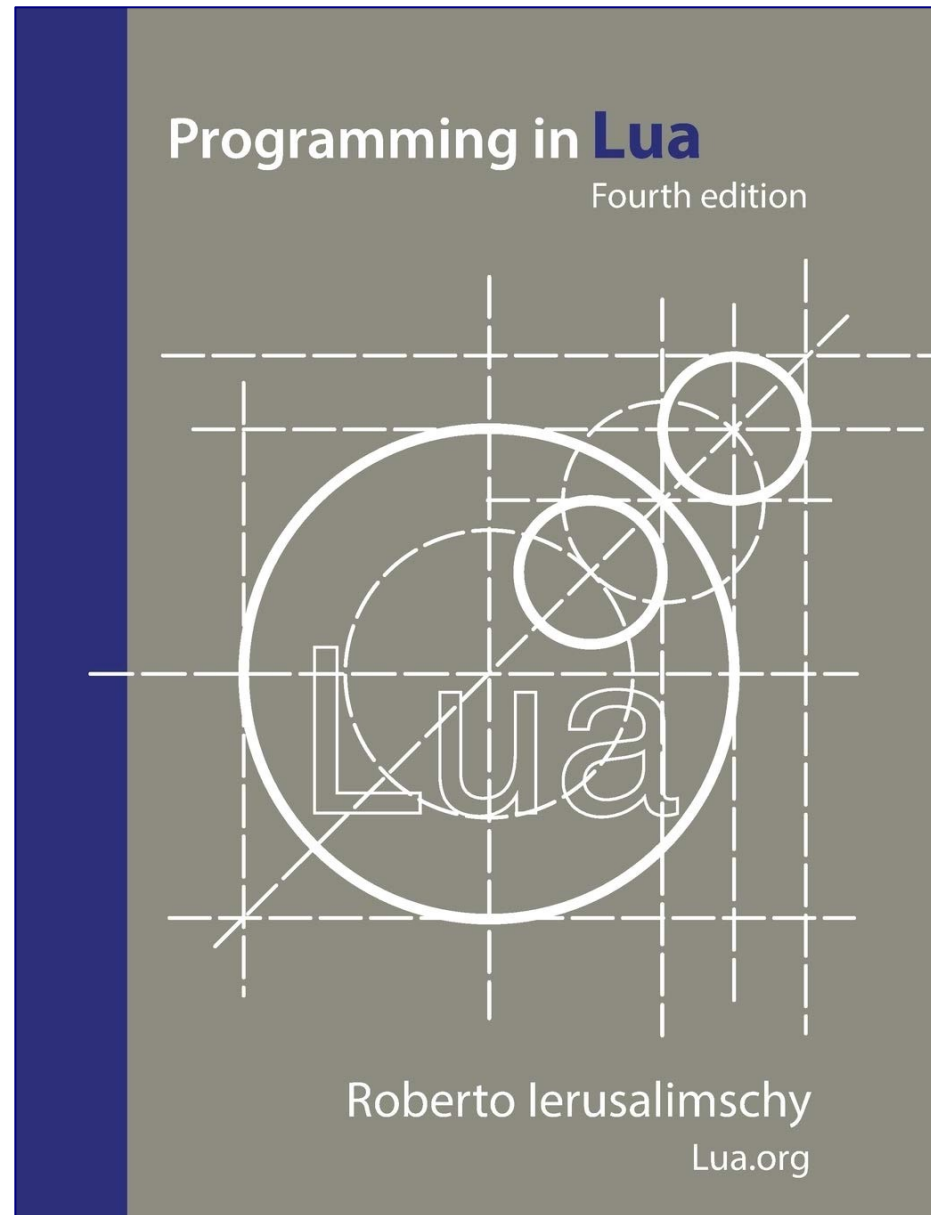
2015

Programando em Lua



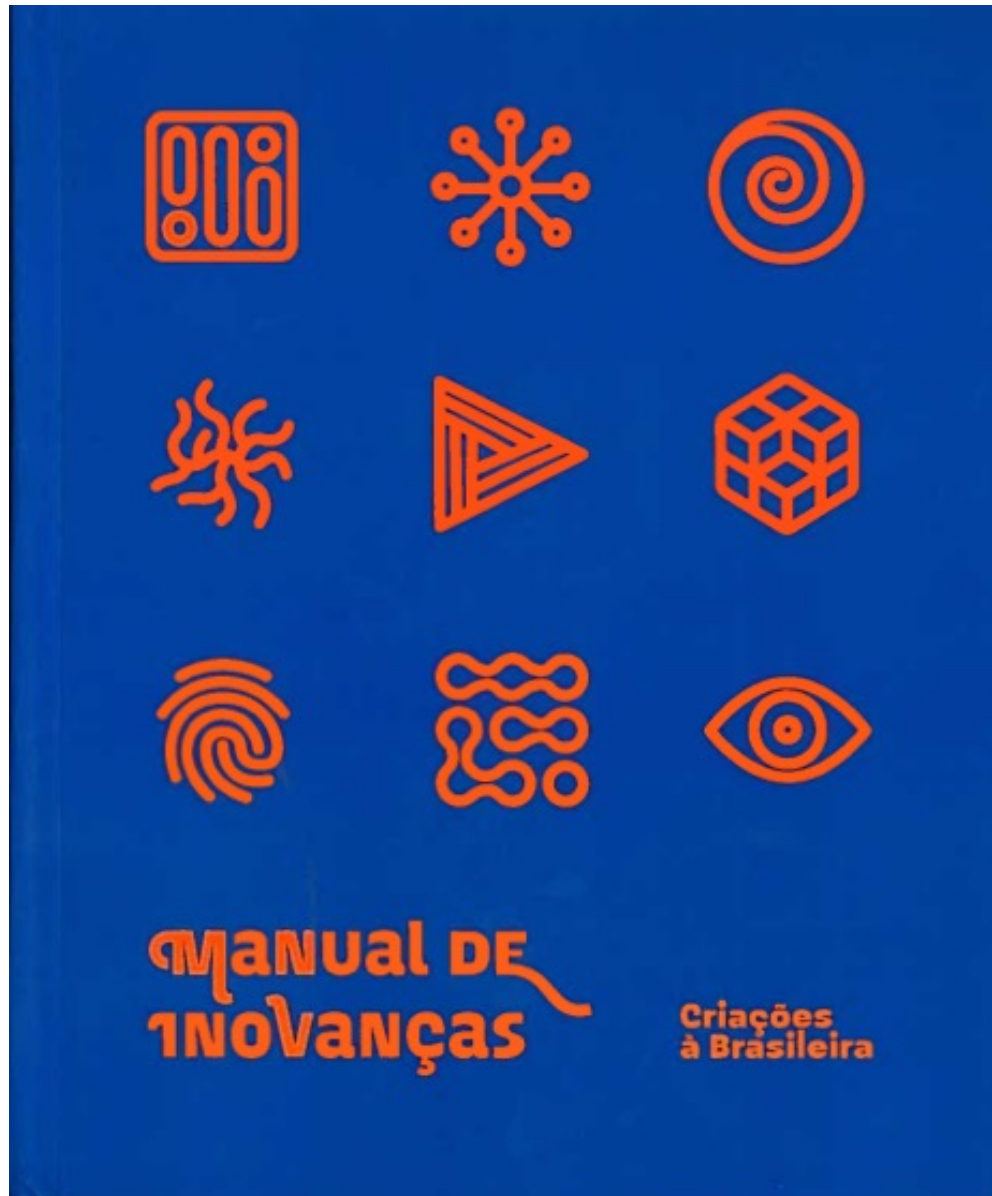
2016

Programming in Lua – 4ed



2017

Inovações – Criações à Brasileira



2017

Inovações – Criações à Brasileira



video





ERRÂNCIAS / CRIAÇÃO 10 / RJ

LINGUAGEM BRASILEIRA EM VIDEOGAMES INTERNACIONAIS – LUA



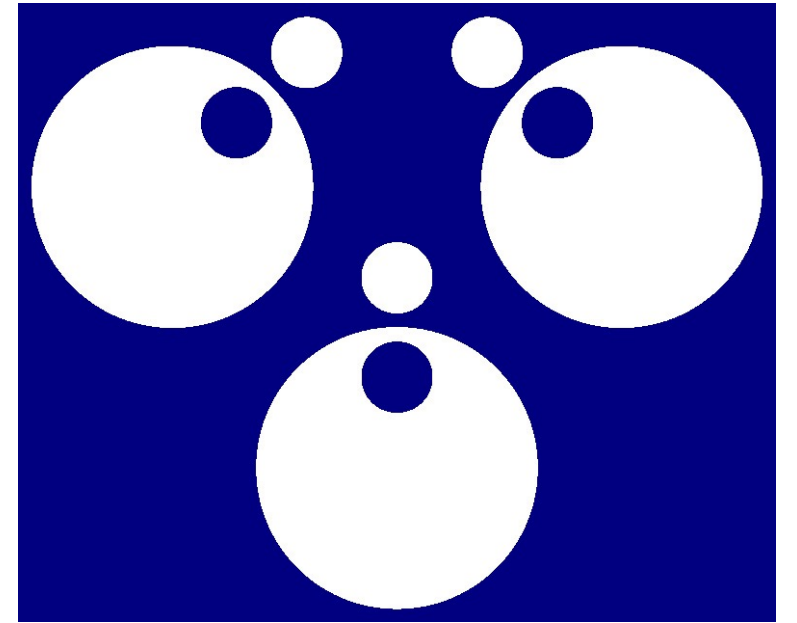
INVENÇÃO AJUDA QUEM SE INTERESSA POR PROGRAMAÇÃO

O que Angry Birds, World of Warcraft, Lego Universe e Star Wars – Empire at War têm em comum? Todos esses videogames trazem em seu DNA uma linguagem de programação desenvolvida no Rio de Janeiro e presente em inúmeros projetos digitais ao redor do mundo: Lua. Essa linguagem é usada, por exemplo, no programa Adobe Photoshop Lightroom, no Ginga (sistema operacional brasileiro para TVs interativas), em roteadores da Cisco e no painel computadorizado dos carros da Volvo e da Mercedes.

Quem poderia prever tamanha popularidade quando esse projeto começou a ser desenvolvido, em 1993? Vivíamos os primórdios da internet e, no mundo dos videogames, Sonic e Mortal Kombat eram grandes novidades. Nessa época, o desafio dos cientistas do laboratório Tecgraf, da Pontifícia Universidade Católica (PUC-Rio) era criar uma linguagem mais simples para projetos de uma empresa petrolífera.

2018

Annual Meeting of TeX Users Group



LuaTeX

DOI:10.1145/3186277

Simplicity, small size, portability, and embeddability set Lua apart from other scripting languages.

BY ROBERTO IERUSALIMSKY, LUIZ HENRIQUE DE FIGUEIREDO, AND WALDEMAR CELES

A Look at the Design of Lua

LUA IS A scripting language developed at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio) that has come to be the leading scripting language for video games worldwide.^{3,7} It is also used extensively in embedded devices like set-top boxes and TVs and in other applications like Adobe Photoshop Lightroom and Wikipedia.¹⁴ Its first version was released in 1993. The current version, Lua 5.3, was released in 2015.

Though mainly a procedural language, Lua lends itself to several other paradigms, including object-oriented programming, functional programming, and data-driven programming.⁹ It also offers good support for data description, in the style of JavaScript and JSON. Data description was indeed one of our main motivations for creating Lua, some years before the appearance of XML and JavaScript.

Our motto in the design of Lua has always been “mechanisms instead of policies.” By policy, we mean a methodical way of using existing mechanisms to

build a new abstraction. Encapsulation in the C language provides a good example of a policy. The ISO C specification offers no mechanism for modules or interfaces.⁹ Nevertheless, C programmers leverage existing mechanisms (such as file inclusion and external declarations) to achieve those abstractions. On top of such basic mechanisms provided by the C language, policy adds several rules (such as “all global functions should have a prototype in a header file” and “header files should not define objects, only declare them”). Many programmers do not know these rules (and the policy as a whole) are not part of the C language.

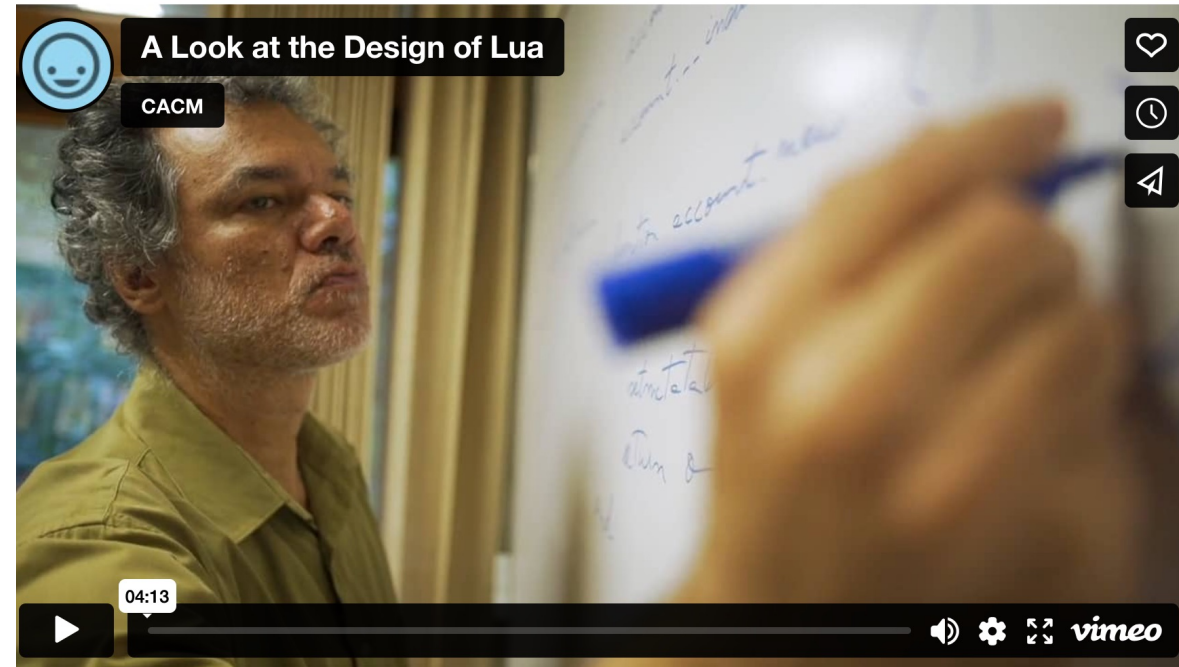
Accordingly, in the design of Lua, we have replaced addition of many different features by creating instead only a few mechanisms that allow programmers to implement such features themselves.⁵ The motto leads to a design that is economical in concepts. Lua offers exactly one general mechanism for each major aspect of programming: tables for data; functions for abstraction; and coroutines for control. On top of these building blocks, programmers implement several other features, including modules, objects, and environments, with the aid of minimal additions (such as syntactic sugar) to the language. Here, we look at how this motto has worked out in the design of Lua.

Design Goals

Like other scripting languages, Lua has dynamic types, dynamic data structures, garbage collection, and an eval-like functionality. Consider Lua’s particular set of goals:

> key insights

- What sets Lua apart from other scripting languages is its particular set of goals: simplicity, small size, portability, and embeddability.
- The entire implementation of Lua has 25,000 lines of C code; the binary for 64-bit Linux has 200k bytes.
- Since its inception, Lua was designed to interoperate with other languages.



video

2022

Medalha Pedro Ernesto





the programming language

Lua

30 years