

Programming Fundamentals

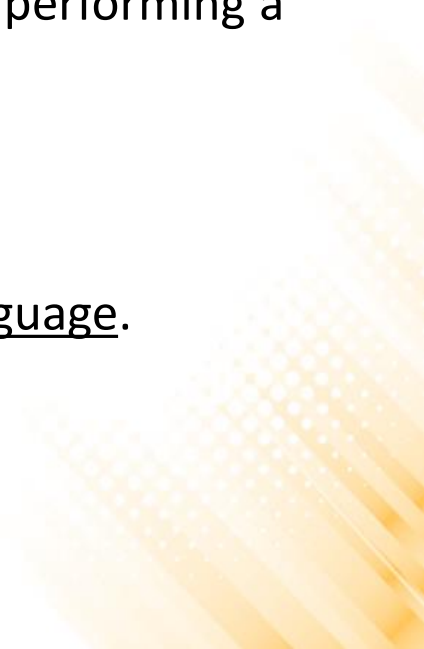
General Course Information

Edirlei Soares de Lima

<edirlei.lima@universidadeeuropeia.pt>



Programming Fundamentals

- **What is computer programming?**
 - The process of writing, testing and maintaining a computer program for accomplishing a specific task.
 - **What is a computer program?**
 - A sequence of instructions that automate the process of performing a task for solving a specific problem.
 - **How to write a computer program?**
 - A computer program is written using a programming language.
- 

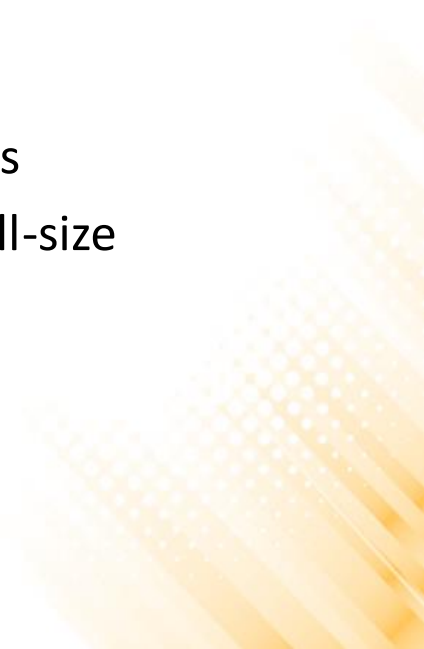
Programming Fundamentals

- **What is a programming language?**
 - A programming language is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks.
 - Examples of programming languages: C, C++, C#, Java, JavaScript, HTML, PHP, Python, Lua, Processing, Ruby, Objective-C, Prolog, Go, ...
- **What is the best programming language?**
 - There is no "best programming language". It depends on the goals of the application, platform, programmer's skills, etc.

Programming Fundamentals


- Professor: Edirlei Soares de Lima
 - Education:
 - B.Sc. in Computer Science – UnC
 - M.Sc. in Computer Science – UFSM
 - Ph.D. in Computer Science – PUC-Rio
 - Teaching Experience: PUC-Rio, UNIRIO, UERJ, IADE-UE
 - Game Experience:
 - Game Engines: RPG Builder, 3D Game Builder (<http://www.3dgamebuilder.com.br/>);
 - Research Projects: most are related with Logtell (<http://www.icad.puc-rio.br/~logtell/>);
 - Games: Krimson (Best Game Award at SBGames 2010 – Indie Game Development Festival), and several other prototype games.
 - More Information: <http://www.inf.puc-rio.br/~elima/>

Programming Fundamentals


- Games & Apps Development:
 - Study the fundamentals of computer programming in a game development context.
 - Learning Outcomes:
 1. Understand the concepts of algorithm and program
 2. Construct algorithms to solve specific problems
 3. Implement simple graphical and interactive applications
 4. Apply the concepts learned in the development of small-size computer games
- 

Programming Fundamentals

- Module Content:

1. Introduction to programming: concepts of algorithm and program
 2. Introduction to Lua: variables, data types, operators and functions
 3. Introduction to computer graphics: coordinates, color, transparency, and Löve 2D
 4. Conditional statements and user interaction
 5. Loops and images
 6. Vectors, physics and collision detection
 7. Arrays, matrices, animations and level representations
 8. Music and audio effects
 9. Introduction to artificial intelligence
- 

Method

- Project-Based Learning:
 - Learn by doing;
 - Teamwork;
 - "Large" project;
 - Active and experiential learning:
 - Theoretical concepts;
 - Practical examples;
 - Implementation exercises;
 - Programming Language: Lua
- 

Evaluation

- Continuous Assessment (bipartite):
 - [60%] Intermediate assessment:
 - [20%] Individual exercises on the concepts learned;
 - [20%] Project with Math, Physics and Games I;
 - [60%] Two intermediate deliveries of the team project (within the semester's PBL team project).
 - [40%] End of term assessment:
 - [100%] Final delivery of the team project (within the semester's PBL team project) with individual discussion.
- Final Assessment:
 - [100%] Individual project development, delivery, and discussion.

Evaluation

- Intermediate project deliveries:
 - [30%] **2nd delivery:** working prototype;
 - [30%] **3rd delivery:** alpha version;
- Final project delivery:
 - **4th delivery:** final version;

Lua Programming Language

- Lua is a powerful, fast, lightweight, embeddable, free and open-source scripting language designed for general purposes.
- In video game development, Lua is one of the most popular scripting language for game programming.
- Lua was developed by a team of researchers at PUC-Rio (Pontifical Catholic University of Rio de Janeiro) in Brazil.
 - Currently, PUC-Rio has a entire laboratory (LabLua) dedicated for the development and maintenance of the Lua language.

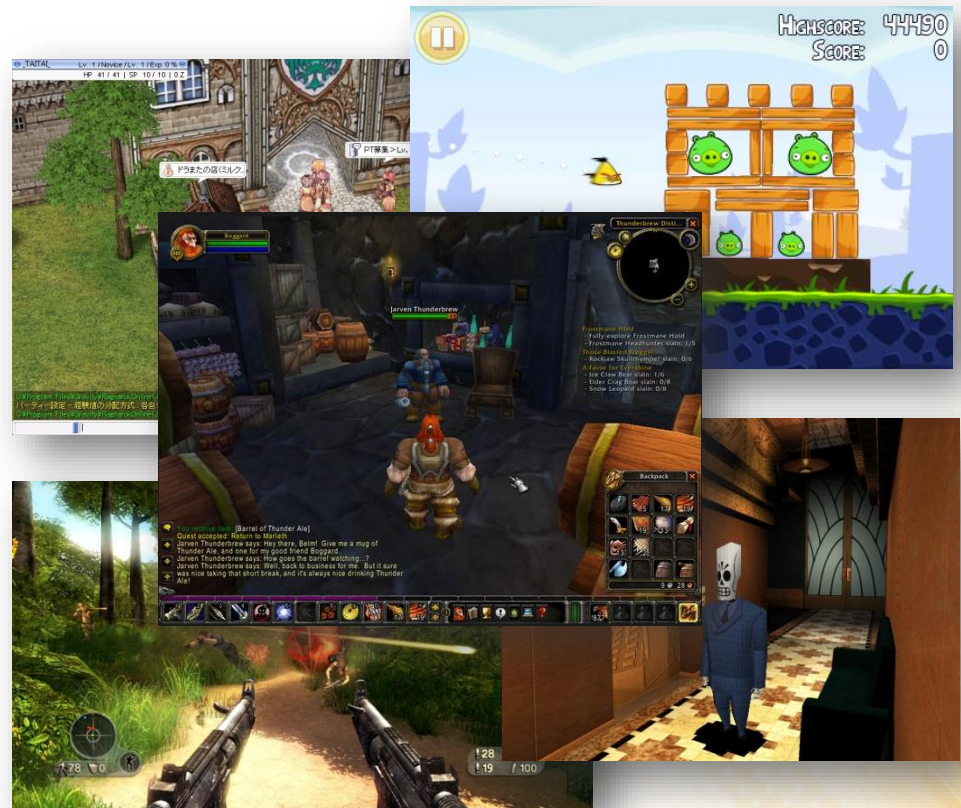


PUC
RIO

Lua Programming Language

- Games that use Lua as scripting language:

- Angry Birds
- Civilization V
- Far Cry
- Grim Fandango
- Ragnarok
- Dota 2
- World of Warcraft



More at: https://en.wikipedia.org/wiki/Category:Lua-scripted_video_games

Löve 2D Framework

- **LÖVE is a framework to create 2D games in Lua.**
 - It comprises a library of functions to draw visual objects, control user interaction, simulate physics, play audio, and everything else a game needs.
 - It's free, open-source and multiplatform (Windows, Linux, Mac OS, Android, iOS, Web Browsers).



Example of Game (Lua + Löve2D)

```
...
function love.draw()
  for i = 1, numStars do
    love.graphics.setColor(255 - stars[i].speed,
                          255 - stars[i].speed/2,
                          150, stars[i].speed*0.9)
    love.graphics.draw(star_image, stars[i].x,
                      stars[i].y, 0, stars[i].speed/255 + 0.55)
  end

  love.graphics.setColor(255, 255, 255)

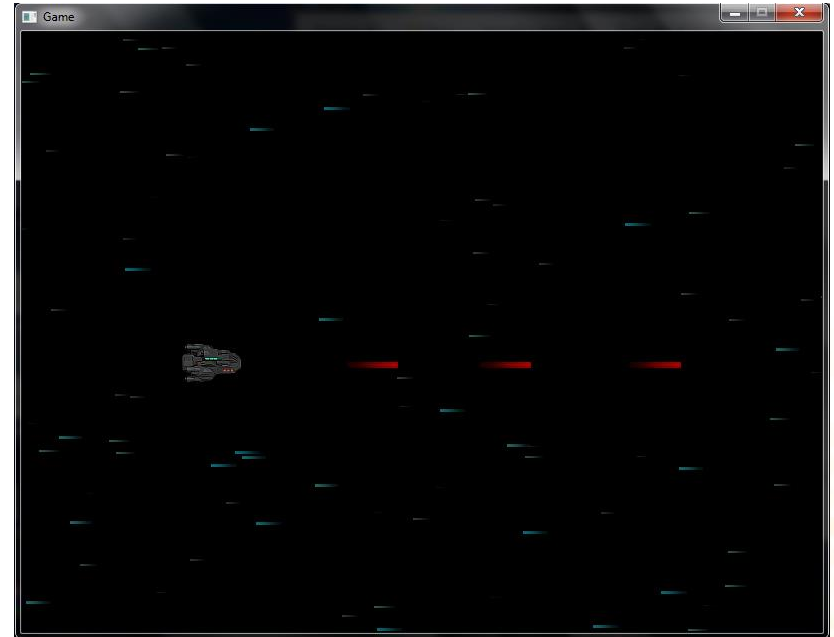
  if spaceship_isBoosted == false then
    love.graphics.draw(spaceship_image_normal,
                      spaceship_x, spaceship_y)
  else
    love.graphics.draw(spaceship_image_boosted,
                      spaceship_x, spaceship_y)
  end

  end

  local totalLaserBeams = table.getn(lasers)

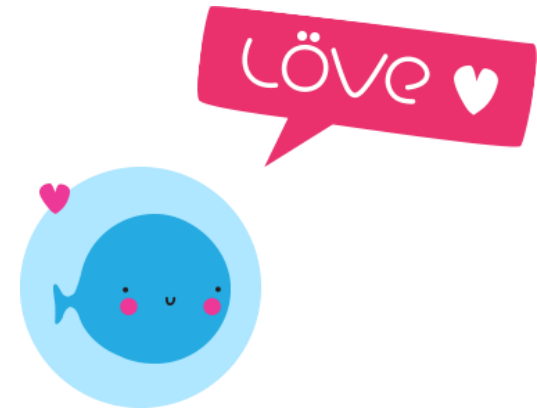
  for i = 1, totalLaserBeams do
    love.graphics.draw(laser_image, lasers[i].x,
                      lasers[i].y)
  end

  end
end
...
```



Software

- Löve: <https://love2d.org/>



- ZeroBrane Studio: <https://studio.zerobrane.com/>

A screenshot of the ZeroBrane Studio IDE. The main window displays a Lua script for a spaceship game. The script includes a function `love.update(dt)` that handles keyboard input for a spaceship. The script is as follows:

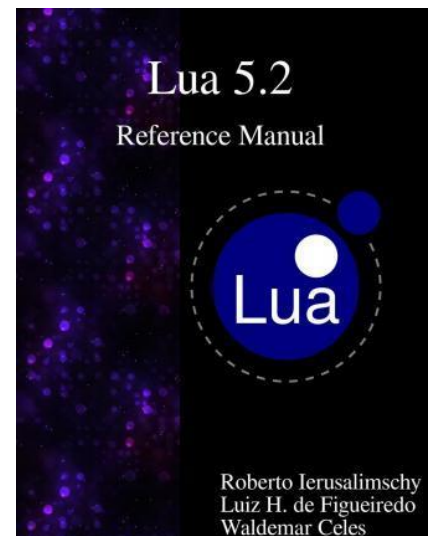
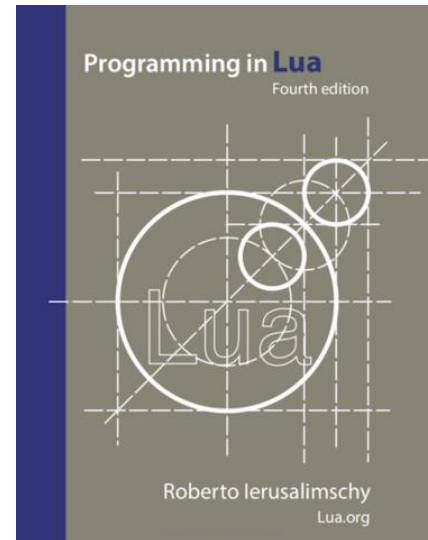
```
function love.update(dt)
    39
    40
    41     if not spaceship.isBoosted then
    42         config.starSpeedFactor = 1
    43     end
    44
    45     if spaceship.isBoosted == false then
    46         if love.keyboard.isDown("left") and (spaceship.x - config.step * dt) >= 0 then
    47             spaceship.x = spaceship.x - config.step * dt
    48         end
    49
    50         if love.keyboard.isDown("right") and (spaceship.x + config.step * dt) < 740 then
    51             spaceship.x = spaceship.x + config.step * dt
    52         end
    53     end
    54
    55     if love.keyboard.isDown("up") and (spaceship.y - config.step * dt) >= 0 then
    56         spaceship.y = spaceship.y - config.step * dt
    57     end
end
```

The bottom of the window shows the output console with the following text:

```
Program starting as "C:\Program Files\love\love.exe" "D:\Dropbox\Aulas\GameProg\Code\Space_Game".
Program 'love.exe' started in 'D:\Dropbox\Aulas\GameProg\Code\Space_Game' (pid: 18836).
Program completed in 12.93 seconds (pid: 18836).
```

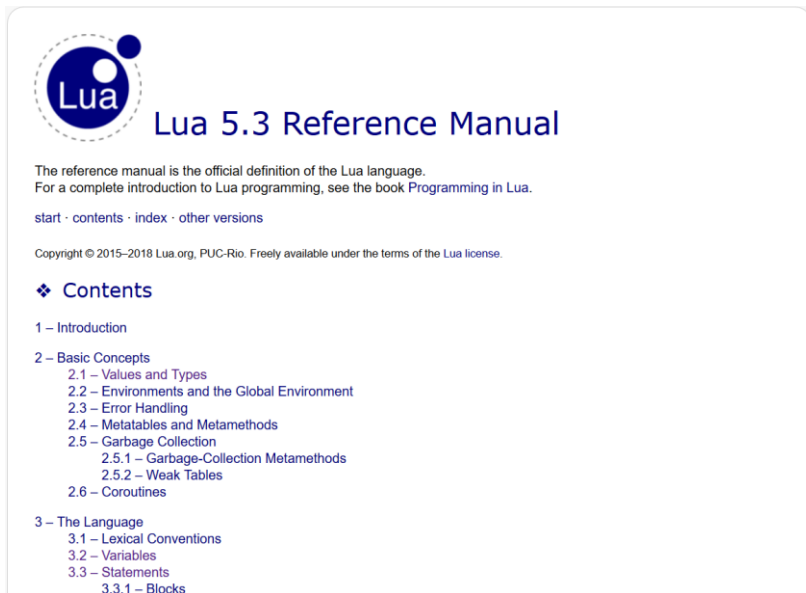
Bibliography

- Ierusalimschy, R. (2016). **Programming in Lua, Fourth Edition**. Lua.Org. ISBN: 978-8590379867
- Ierusalimschy, R., Figueiredo, L. H., Celes, W. (2015). **Lua 5.2 Reference Manual**. Samurai Media Limited. ISBN: 978-9888381227




Web Resources

- Lua 5.3 Reference Manual: <https://www.lua.org/manual/5.3/>
- Löve Documentation: https://love2d.org/wiki/Main_Page



The screenshot shows the title page of the Lua 5.3 Reference Manual. It features the Lua logo (a blue circle with a white dot and the word 'Lua' inside) and the title 'Lua 5.3 Reference Manual'. Below the title, there is a short introduction: 'The reference manual is the official definition of the Lua language. For a complete introduction to Lua programming, see the book Programming in Lua.' There are links for 'start', 'contents', 'index', and 'other versions'. At the bottom, there is a 'Contents' section with a diamond icon, listing three main parts: 1 - Introduction, 2 - Basic Concepts (with sub-sections 2.1 to 2.6), and 3 - The Language (with sub-sections 3.1 to 3.3.1).

 Lua 5.3 Reference Manual

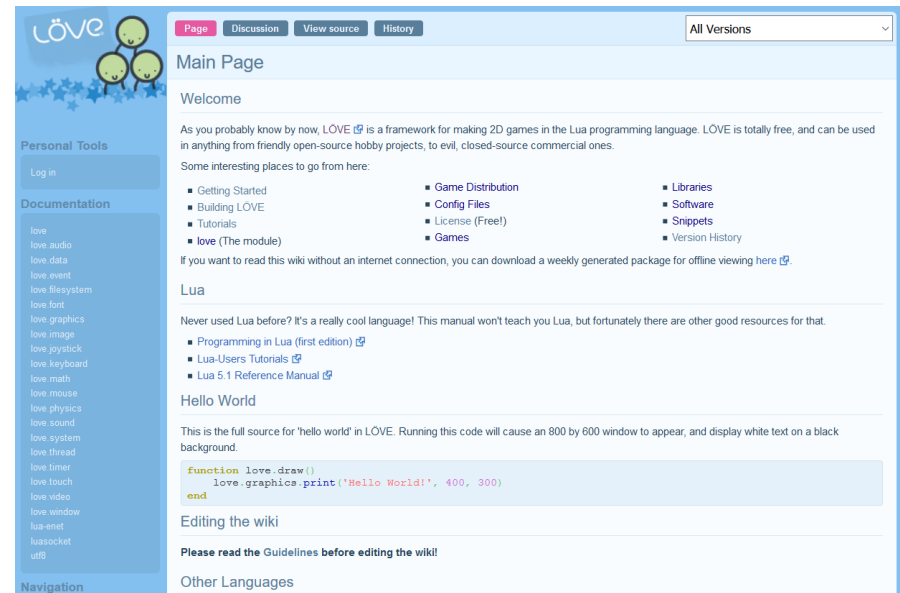
The reference manual is the official definition of the Lua language.
For a complete introduction to Lua programming, see the book *Programming in Lua*.

[start](#) · [contents](#) · [index](#) · [other versions](#)


Copyright © 2015–2018 Lua.org, PUC-Rio. Freely available under the terms of the Lua license.

◆ Contents

- 1 – Introduction
- 2 – Basic Concepts
 - 2.1 – Values and Types
 - 2.2 – Environments and the Global Environment
 - 2.3 – Error Handling
 - 2.4 – Metatables and Metamethods
 - 2.5 – Garbage Collection
 - 2.5.1 – Garbage-Collection Metamethods
 - 2.5.2 – Weak Tables
 - 2.6 – Coroutines
- 3 – The Language
 - 3.1 – Lexical Conventions
 - 3.2 – Variables
 - 3.3 – Statements
 - 3.3.1 – Blocks



The screenshot shows the 'Main Page' of the Löve Documentation. It has a blue header with the 'love' logo (three green circles) and navigation tabs for 'Page', 'Discussion', 'View source', and 'History'. Below the header, there is a 'Welcome' section with a paragraph about Löve being a free framework for making 2D games in Lua. There is a list of 'Some interesting places to go from here:' with links to 'Getting Started', 'Building LOVE', 'Tutorials', 'love (The module)', 'Game Distribution', 'Config Files', 'License (Free!)', 'Games', 'Libraries', 'Software', 'Snippets', and 'Version History'. There is also a link to download a weekly generated package for offline viewing. Below that, there is a section for 'Lua' with a paragraph about it being a cool language and links to 'Programming in Lua (first edition)', 'Lua-Users Tutorials', and 'Lua 5.1 Reference Manual'. There is a 'Hello World' section with a code block showing a Lua function that prints 'Hello World!' in a window. At the bottom, there is a section for 'Editing the wiki' with a link to 'Please read the Guidelines before editing the wiki!' and a section for 'Other Languages'.

 [Page](#) [Discussion](#) [View source](#) [History](#) All Versions

Main Page

Welcome

As you probably know by now, **LOVE** is a framework for making 2D games in the Lua programming language. LOVE is totally free, and can be used in anything from friendly open-source hobby projects, to evil, closed-source commercial ones.

Some interesting places to go from here:

- Getting Started
- Building LOVE
- Tutorials
- love (The module)
- Game Distribution
- Config Files
- License (Free!)
- Games
- Libraries
- Software
- Snippets
- Version History

If you want to read this wiki without an internet connection, you can download a weekly generated package for offline viewing [here](#).

Lua

Never used Lua before? It's a really cool language! This manual won't teach you Lua, but fortunately there are other good resources for that.

- [Programming in Lua \(first edition\)](#)
- [Lua-Users Tutorials](#)
- [Lua 5.1 Reference Manual](#)

Hello World

This is the full source for 'hello world' in LOVE. Running this code will cause an 800 by 600 window to appear, and display white text on a black background.

```
function love.draw()  
    love.graphics.print("Hello World!", 400, 300)  
end
```

Editing the wiki

Please read the [Guidelines](#) before editing the wiki!

Other Languages

Programming Fundamentals

- Blackboard (Programming Fundamentals):
 - <https://europeia.blackboard.com/>
- Course Webpage:
 - <http://www.inf.puc-rio.br/~elima/gameprog/>
- Contact:
 - edirlei.lima@universidadeeuropeia.pt
 - edirlei.slima@gmail.com