Exploring Lua

Exploring Lua

- Why & What
- Tour of Lua
- Lua and iOS
- Study: Wax Framework
- Findings & Conclusion
- Q&A

Why?

- More efficient way to write apps?
- Expand your toolbox
- New Language, New Perspective
- Change the way you build/compose software

What is Lua?

- Fast, lightweight, embeddable scripting language
- Highly portable - ANSI C
- Interpreted, dynamically typed, incremental GC
- Small, simple & powerful language
- Lua - "Moon" in Portuguese

Language Tour

- Flavour of the Syntax
- Functions
- Tables
- OOP
- Embedding Lua

Lua Syntax

```
print("hello world")

function factorial(n)
  if n == 0 then
    return 1
  else
    return n * factorial(n - 1)
  end
end

print("factorial(20) is: " .. factorial(20))
```

Functions

```
function timesBy(n)
    return  function(x)
                return x * n
            end
end

local twoX = timesBy(2)
local threeX = timesBy(3)

for i = 1, 10 do
    print(i .. " | " .. twoX(i) .. " | " .. threeX(i))
end
```

Tables

```
-- table as an array
names = {'mark', 'jenny'}
for i = 1, #names do
    print(names[i])
end


-- table as a record
m = { name = "Mark", title = "Dr", age = 29 }
```

Tables contd.

```
-- table as a namespace
Point = {}

Point.new = function (x, y)
  return {x = x, y = y}
end

Point.set_x = function (point, x)
  point.x = x
end
```

OOP

- Tables not Objects are the primitive of Lua
- OOP implemented using Tables
- Lua provides some syntactic sugar
- Implement OOP using Lua facilities
  - Metatable facility

OOP Example

```
Vector = {}                  -- Create a table to hold the class methods
function Vector:new(x, y, z)  -- The constructor
  local object = { x = x, y = y, z = z }
  setmetatable(object, { __index = Vector })  -- Inheritance
  return object
end
function Vector:magnitude()   -- Another member function
  -- Reference the implicit object using self
  return math.sqrt(self.x^2 + self.y^2 + self.z^2)
end

vec = Vector:new(0, 1, 0)      -- Create a vector
print(vec:magnitude())         -- Call a member function using ":"
print(vec.x)                   -- Access a member variable using "."
```

Embedding Lua

WORLD OF WARCRAFT

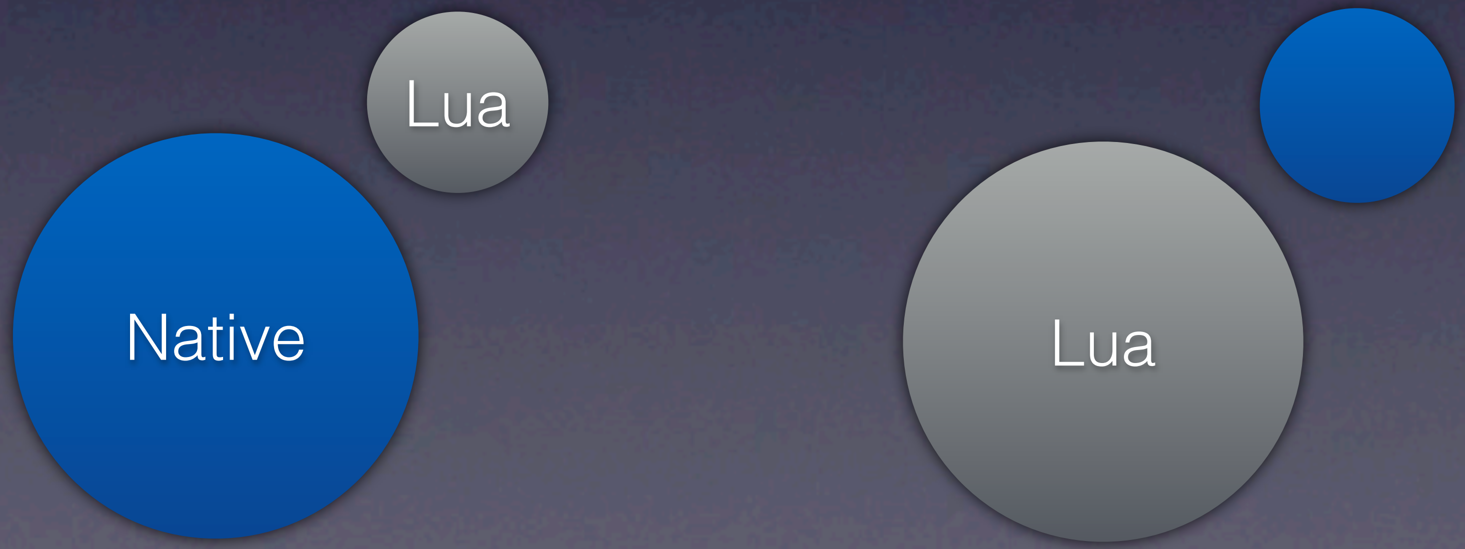
Tap Tap Revenge

Embedding

- Data files & configuration
- Scripting and plugins
- Embedding vs Extending

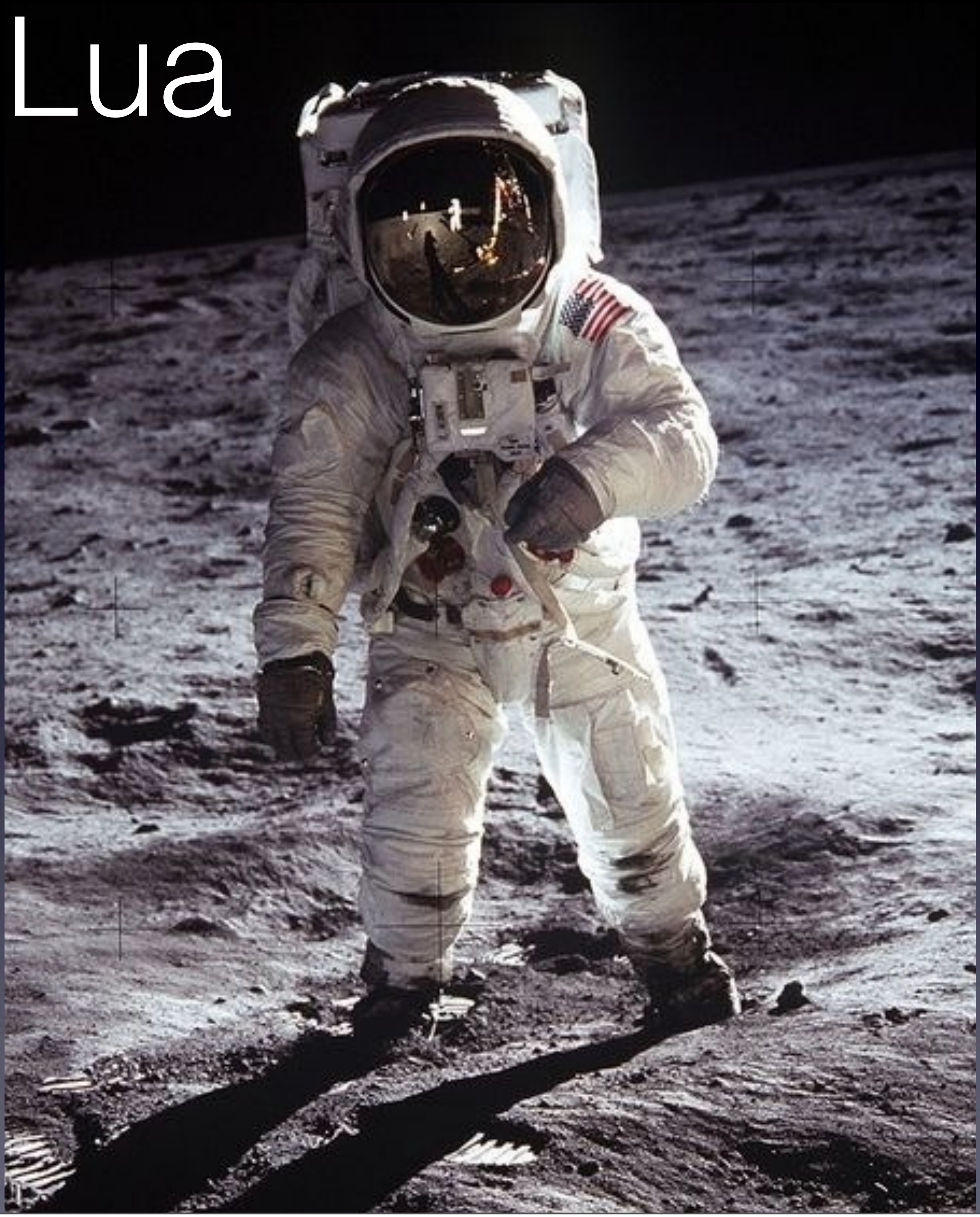
Embedding

- Data files & configuration
- Scripting and plugins
- Embedding vs Extending



Exploring Lua

- Why & What
- Tour of Lua
- **Lua and iOS**
- Study: Wax Framework
- Findings & Conclusion
- Q&A



Lua and iOS

- Proven history in embedded systems
- Being used in iOS apps
- Flexible enough to allow multiple approaches

Lua and iOS

- Embedded in Games and Apps
- 3rd Party Tools: Ansca's Corona
- Build complete Cocoa apps in Lua

Wax Framework

- A Lua bridge to Objective-C
- Write Cocoa Apps in Lua
- Open source (non-viral)
- http://github.com/probablycorey/wax

Wax: Objects & Messages

```
local view = UIView:initWithFrame(CGRect(0, 0, 320, 100))

-- use a colon when sending a message to Obj-C Object
view:setBackgroundColor(UIColor:redColor())


-- multiple arguments...
local sa = UIApplication:sharedApplication()
sa:setStatusBarHidden_animated(true, false)
```

Wax: Data Types

```
-- Automatically converts array/string/dictionary objects to NSArray,
-- NSString and NSDictionary objects (and vice-versa)

images = {"myFace.png", "yourFace.png", "theirFace.png"}
imageView = UIImageView:initWithFrame(CGRect(0, 0, 320, 460))
imageView:setAnimationImages(images)
```

Wax : View Controller

```
-- Creates an Obj-C class: MyController

waxClass{"MyController", UIViewController}

function init()
  -- to call a method on super, simply use self.super
  self.super:initWithNibName_bundle("MyControllerView.xib", nil)
  return self
end

function viewDidLoad()
  -- Do all your other stuff here
end
```

Wax: Initial Findings

- Learning Curve.
- Powerful. Less Code.
- Small community.
- Prototyping -> App
- Credible and useful development tool

Wax: Annoyances

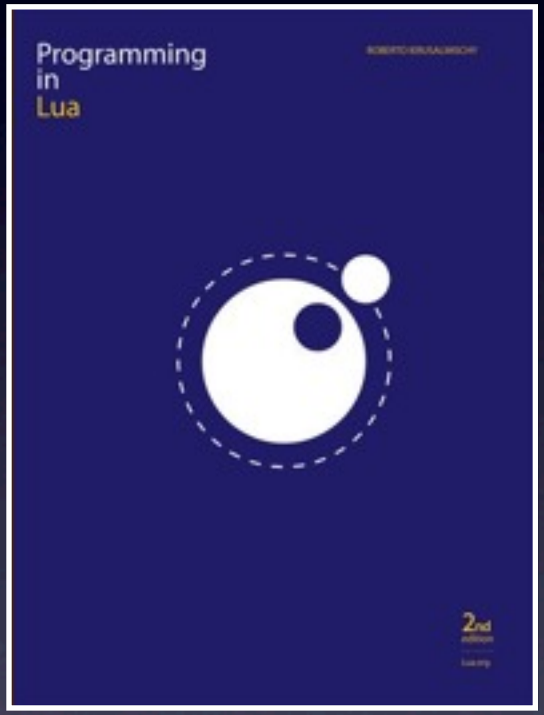
- Syntax issues: Doesn't use properties
- IB support
- No Xcode 4 project template (yet)
- Learning curve for experienced iOS developers
- Documentation and examples
- Debugging

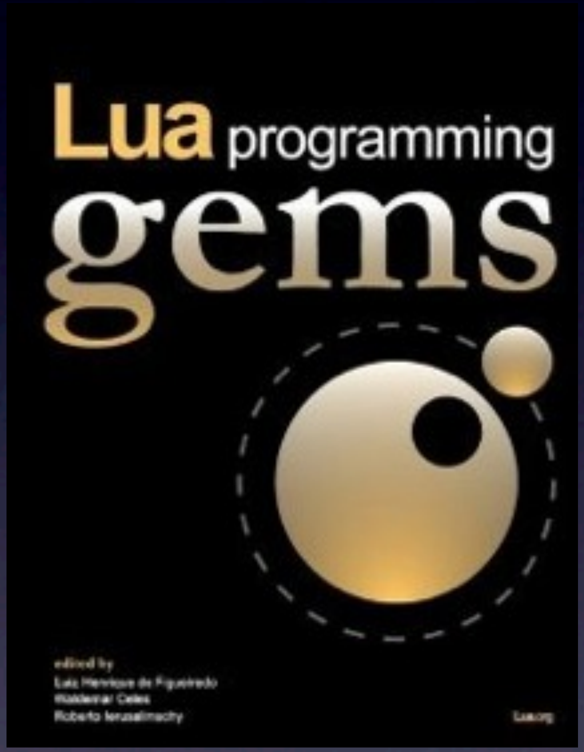
Conclusion

- Lua is a great addition to toolbox
- Used in a lot of places
- Wax - lot of potential
- More to explore...
  - Lua. Wax.
  - Nu, Gambit Scheme...

Learn More

- http://lua.org
- http://lua-users.org
- Programming in Lua
- Lua Programming Gems





Credits

http://www.flickr.com/photos/penguinbush/2768719983/sizes/o/in/photostream/

http://www.flickr.com/photos/nasacommons/5052744678/sizes/l/in/photostream/

http://www.flickr.com/photos/nasacommons/5136519916/

Questions

krishna@logiccolony.com

@kenshi