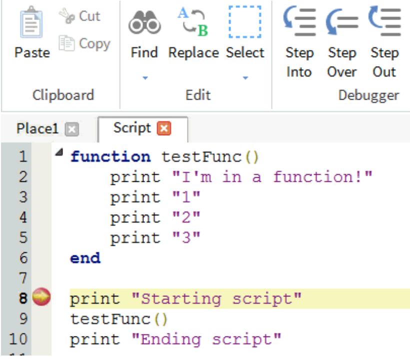
The Lua Programming Language

Jonathan Carter
Keeling Wood
Siddhartha Das
Justin Szajek
Chris Cokus

What is Lua?

- Free, embeddable scripting language
- Popular in the game industry
- Created to support a wide variety of application customization
- Ideal for, “configuration and scripting”
- It can run on any device with a standard C compiler



```lua
function testFunc()
    print "I'm in a function!"
    print "1"
    print "2"
    print "3"
end

print "Starting script"
testFunc()
print "Ending script"
```

History

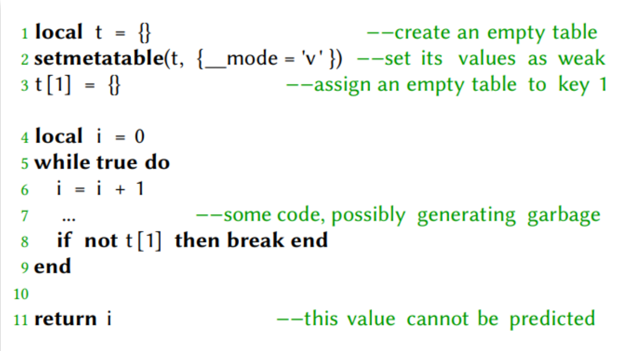
- Born in 1993, and (still) raised by a team at the PUC-Rio
    - Started by the Computer Graphics Technology Group of PUC-Rio (TeCGraf),
    - Moved to LabLua, a laboratory of their CS dept
- DEL (Data Entry Language) + Sol (Simple Object Language) → Lua
- Made with “meta-mechanisms” rather than features

Type Language

- Lua is a dynamically typed language
  - All values are first-class values
- Lua has 8 basic types:
  - nil, boolean, number, string, function, userdata, thread, and table
  - Similar to python, has a type function to retrieve the type of a variable
- String:
  - Automatic conversion between number and string at run-time
  - Number to string
    - Integer -> w/o exponent or decimal
    - Others -> follows C format specifier rules of “%g”

Memory Management

- Automatic memory management (garbage collector or GC)
  - Uses i**ncremental mark** and **sweep collector**
  - Consistency of GC with **garbage collector pause**
  - Aggressiveness of GC with **garbage collector step multiplier**
- Has a system of *weak*/*strong* references
  - All references to an object are weak -> can be collected
- Concept of *weak tables*
  - All stored references are weak -> eventually be collected
  - Cleans corresponding values in table
  - Strings aren’t considered “collectible”
- LuaSafe (tool)



```
local t = {}                          --create an empty table
setmetatable(t, {__mode = 'v' })      --set its values as weak
t[1] = {}                             --assign an empty table to key 1

local i = 0
while true do
  i = i + 1
  ...                     --some code, possibly generating garbage
  if not t[1] then break end
end

return i                              --this value cannot be predicted
```

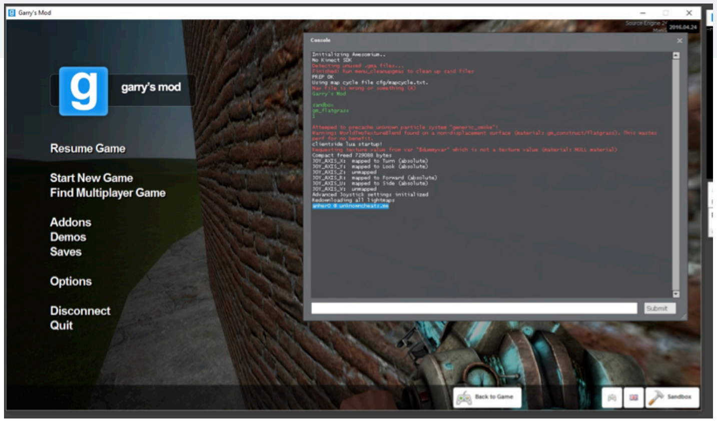
Figure 1: A non-deterministic program using a weak table.

Features/Uses

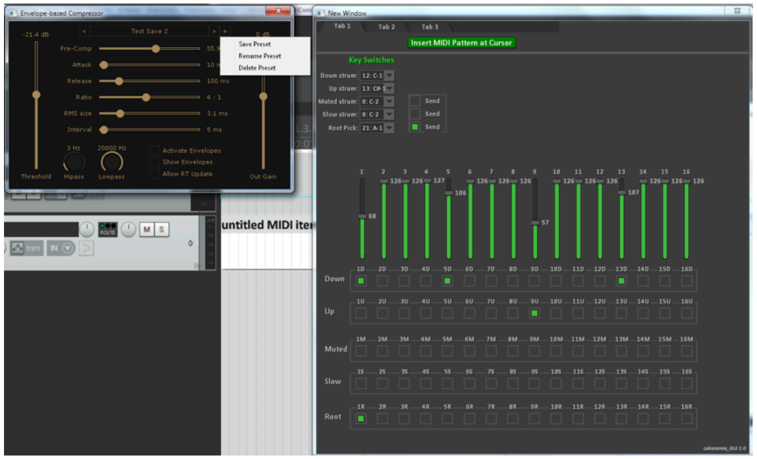
- Optimal for handling simple tasks
- Easy to embed; high compatibility, extensible
- Notable runtime (faster than python)
    - Ex: Short circuiting for ‘and’/‘or’ evaluations
- Simple syntax
    - No type declaration
    - No ‘int’ type; all number values are doubles (64-bit, 52 for int value)
    - ‘..’ operator for string concatenation (“hello ” .. “world” == “hello world”)
    - Uses ‘~=’ for inequality, not ‘!=’
    - Functions can return multiple values (a, b, c = (function: ‘return 1, 2, 3’): makes a=1, b=2, c=3)
- Style of language lends itself to compact code, simple execution, and fast runtime

Real World Examples

- CRYENGINE (Engine) -> User Scripts
- Adobe Lightroom (App) -> UI
- REAPER (DAW) -> extending functionality
- Garry's Mod + Payday 2 + etc -> memes/mods



```
Initializing Awesomium..
No Kinect SDK
Detecting unused .gma files...
Finished! Run menu_cleanupgmas to clean up said files
PREP OK
Using map cycle file cfg/mapcycle.txt.
Nav File is wrong or something (4)
Garry's Mod

sandbox
gm_flatgrass
1

Attemped to precache unknown particle system "generic_smoke"!
Warning: WorldTwoTextureBlend found on a non-displacement surface (material: gm_construct/flatgrass). This wastes perf for no benefit.
clientside lua startup!
Requesting texture value from var "$dummyvar" which is not a texture value (material: NULL material)
Compact freed 729088 bytes
JOY_AXIS_X:  mapped to Turn (absolute)
JOY_AXIS_Y:  mapped to Look (absolute)
JOY_AXIS_Z:  unmapped
JOY_AXIS_R:  mapped to Forward (absolute)
JOY_AXIS_U:  mapped to Side (absolute)
JOY_AXIS_V:  unmapped
Advanced Joystick settings initialized
Redownloading all lightmaps
anher0 @ unknowncheats.me
```



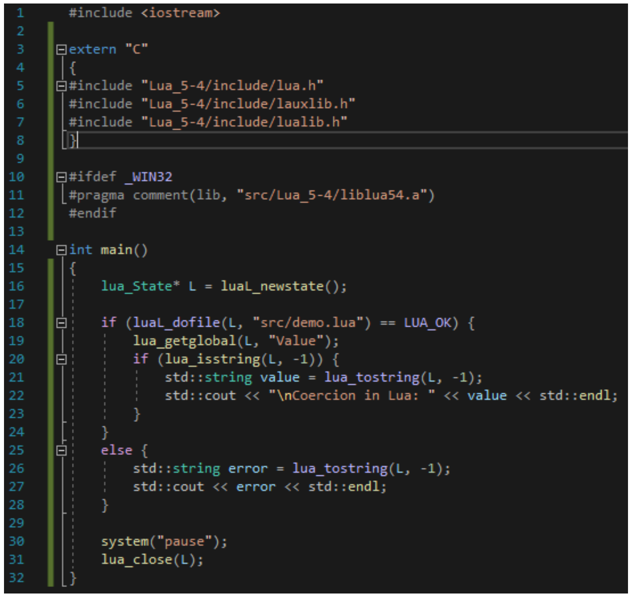
-21.4 dB
Pre-Comp 55.9
Attack 10 m
Release 100 ms
Ratio 4 : 1
RMS size 3.1 ms
Interval 5 ms
3 Hz
20000 Hz
0 dB
□ Activate Envelopes
□ Show Envelopes
□ Allow RT Update

Down strum: 12: C-1
Up strum: 13: C#-1
Muted strum: 0: C-2 □ Send
Slow strum: 0: C-2 □ Send
Root Pick: 21: A-1 ☑ Send

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
68	126	126	127	106	126	126	126	57	126	126	126	107	126	126	126

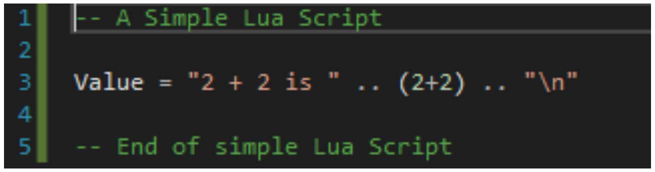
Example of Embedding Lua in C++

C++



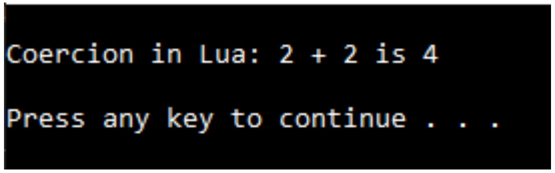
```cpp
#include <iostream>

extern "C"
{
#include "Lua_5-4/include/lua.h"
#include "Lua_5-4/include/lauxlib.h"
#include "Lua_5-4/include/lualib.h"
}

#ifdef _WIN32
#pragma comment(lib, "src/Lua_5-4/liblua54.a")
#endif

int main()
{
    lua_State* L = luaL_newstate();

    if (luaL_dofile(L, "src/demo.lua") == LUA_OK) {
        lua_getglobal(L, "Value");
        if (lua_isstring(L, -1)) {
            std::string value = lua_tostring(L, -1);
            std::cout << "\nCoercion in Lua: " << value << std::endl;
        }
    }
    else {
        std::string error = lua_tostring(L, -1);
        std::cout << error << std::endl;
    }

    system("pause");
    lua_close(L);
}
```

Lua Script



```lua
-- A Simple Lua Script

Value = "2 + 2 is " .. (2+2) .. "\n"

-- End of simple Lua Script
```

Output



```
Coercion in Lua: 2 + 2 is 4

Press any key to continue . . .
```

Discussion

- Which applications (games, productivity, etc) do you know that use Lua scripts?
  - World of Warcraft - Online multiplayer role-playing game
  - Angry Birds
  - Firefox web browser

Links

https://www.tutorialspoint.com/lua/lua_overview.htm

http://www.lua.org/pil/1.html

https://www.lua.org/about.html

https://arxiv.org/pdf/2005.13057.pdf

https://www.lua.org/pil/17.html