Speed of Lua embedded in C++

Patrick Schuur

# Introduction

When you want users to make additional content for your software like HUD improvements in your MMORPG or functionality in your sandbox game you don’t want them to feel overwhelmed doing it. To prevent this you would make an API that the user can interact with instead of the pure underlying code.

Maybe you would even go as far as making it possible for the user to work in a wholly different (usually interpreted) programming language than the original software was made in. You could do this because the other language is easier for novice programmers to understand or because it makes quick scripting easier because the user doesn’t have to think about the things that would come into play with a compiled language. This is called language embedding.

Tooling like this is often used in big title games like World of Warcraft for addons and in Roblox for game functionality. In both these cases the scripting language Lua is used, Lua is interpreted, dynamically typed, and has automatic memory management.

Embedded languages can also be used during game development to make it easier for designers to implement functionality so programmers can focus on other tasks. In this paper I explore how the scripting language Lua compares in speed to native C++ code and whether the speed difference is significant enough to not use it in game development.

# References

***Programming in Lua (first edition). (n.d.).***[***https://www.lua.org/pil/contents.html***](https://www.lua.org/pil/contents.html)

***javidx9. (2019, March 24). Embedding Lua in C++ #1 [Video]. YouTube.*** [***https://www.youtube.com/watch?v=4l5HdmPoynw***](https://www.youtube.com/watch?v=4l5HdmPoynw)

***Standard library header (C++11) - cppreference.com. (n.d.).*** [***https://en.cppreference.com/w/cpp/header/chrono***](https://en.cppreference.com/w/cpp/header/chrono)

***Iterate through Lua Table. (n.d.). Stack Overflow.*** [***https://stackoverflow.com/questions/6137684/iterate-through-lua-table***](https://stackoverflow.com/questions/6137684/iterate-through-lua-table)

# Method

For my testing I used the chrono time library from the C++ standard template library to make a simple timer to record the time my tests took to complete. I opted to record the time in nanoseconds as many C++ tests returned a duration of 0 because it took too little time.

Text

Description automatically generated

I do the same test multiple times to get an average duration, sometimes however the first and last result can skew the results a lot, so I left them out of the equation.

Text

Description automatically generated

# Palindromes

For my first test I wanted to test the speed of finding all the palindromes up until a million. Even though finding palindromes isn’t very useful for game development it does help with show how fast Lua is at string manipulation, looping and table manipulation.

*Text

Description automatically generated*Text

Description automatically generated

*Function in Lua Function in C++*

Graphical user interface, text

Description automatically generated

*CPP Test*

Text

Description automatically generated

*Lua Test*

I decided to run the test 3 different times: the first I would run 10 times, the second I would run 100 times and the last one 1000 times.

For the graphs I rounded all the values to 3 decimal places to get seconds instead of nanoseconds.

The last test seems to have a significantly higher average for Lua than the other tests, I first thought this had to with me doing other things on my laptop during the test as the test took around 20 minutes, but this doesn’t explain the fact that C++ doesn’t have this issue.

I can however conclude from these tests that Lua is significantly worse for these kinds of tasks although I think that is fine as that isn’t really the purpose of the language.