Project Proposal: MoonScript

Student Name(s): Elena Adame

Proposed System Under Test (SUT): moonscript

Link to SUT Source Code: https://github.com/leafo/moonscript

SUT Size: 21,970 Lines of Code SUT

Description MoonScript is a dynamically typed scripting language that compiles into LUA either run at a later time. It is fully compatible with LUA libraries and and its implementations. I have not worked with LUA, and the only other scripting language I am familiar with is Python. State of SUT MoonScript is 7 years old and is kind of actively developed with 741 commits, the last one being 11 days ago. There are some specifications and documentation (https://moonscript.org/reference/).

Attributes

● Interactive: MoonScript allows programmers to incrementally write code and see code results.

● Modern: MoonScript makes use of the latest programming language features.

● Fast: MoonScript code runs quickly.

Components

● Code Generator: Compiles MoonScript code to LUA, with LPeg (http://www.inf.puc-rio.br/%7Eroberto/lpeg/).

- Destructure Built in: Moonscript can unpack arrays.

Capabilities:

● REPL is interactive: The REPL can check if super types are supposed to be an object or a function.

- Destructure is Modern: Destructuring assignment is a way to quickly extract values from a table by their name or position in array-based tables.

Unit Tested Capabilities(s): Unit tests will be written for the destructure ability. Moonscript claims to be efficient in that you can quickly extract values based on name and not have to loop through items. I plan to test their unique assignment and classes.

Automatically Tested Capabilities(s): I plan to test the code generator for classes. I want to automatically test if the expected Lua output matches the actual Lua output. If there are discrepancies their code generator has problems. I plan to automate this check with another language, maybe java or python.