**Respondent 1:**

* Amateur programmer (3/7)
* Advanced Lua programmer (5/7)
* Games: Binding of Isaac
* Thinks Lua is easy to use and suitable for the game.
* Finds it odd that Lua has no separation between integer and floats.
* Problems gravitate more towards the API rather than anything Lua is providing.
* Lua/game modding was their entry into programming.

**Respondent 2:**

* Advanced programmer (6/7)
* Advanced Lua programmer (6/7)
* Games: Binding of Isaac and One Step from Eden (interestingly, a JRPG TCG hybrid)
* Thinks Lua is well suited to the task for both games (despite the BOI API not being fit for purpose).
* Only knows Python outside of Lua (I would say this puts them at most a 5/7 for their programming expertise in my opinion).
* Lua does not support compound assignment operators (+=, -=, \*=, /=, %=) and increment/decrement operators (i++, i--, ++i, --i).
* Also learnt through modding.

**Respondent 3:**

* Advanced programmer (5/7)
* Expert Lua programmer (7/7)
* Games: World of Warcraft
* Also thinks Lua is well suited to making mods.
* Not much other info – says Lua could do with OOP features, but to be fair to it these are already available (albeit in a strange way).

**Respondent 4:**

* Intermediate programmer (4/7)
* Advanced Lua programmer (5/7)
* Games: Garry’s Mod
* Believes that Lua is a “double-edged sword”. On the one hand, it is fast and compact. On the other hand, when using it, they miss utility methods and other tools that something like C# would have.
* Mentions compound assignment operators.

**Respondent 5:**

* Amateur programmer (3/7)
* Advanced Lua programmer (5/7)
* Games: Binding of Isaac
* Also began learning programming through modding in Lua.
* Does not like using # to find the length of an array as it is “faulty on non-numeric indices”.
* Likes the way if statements are in comparison to using brackets in other languages.
* Likes how it doesn’t require semicolons.
* Wants a map function.
* Also believes that the Binding of Isaac API is not fit for purpose.

**Respondent 6:**

* Advanced programmer (6/7)
* Advanced Lua programmer (5/7)
* Games: Civilization VI, Civilization V
* Think Lua works fine for modding due to its lightweight nature.
* Difficult to debug without using print statements everywhere.
* Wants documentation.

**Respondent 7:**

* Expert programmer (7/7)
* Expert Lua programmer (7/7)
* Games: Garry’s Mod
* Believes Lua is suitable for mod development.
* Wants the following improvements:
  + async/await
  + buffers
  + less ‘snowflake’ syntax for the sake of being different (not sure what exactly they are referring to here since all languages do this to an extent, so we’d have to know what their preferred language is)
  + no continue keyword
  + threading
  + asynchronous IO

**Respondent 8:**

* Advanced programmer (5/7)
* Advanced Lua programmer (5/7)
* Games: Factorio
* Believes Lua is suitable for modding.
* Likes its simplicity.
* Wants function overloading and more typing (I think…)
* *When values are returned from a function in Lua, they must be put into parentheses before indexing. It's not a significant problem, just a slight disturbance of how one may expect the syntax to operate.* – Not entirely sure what they mean by this.
* Would like a tool that assists with the API documentation, i.e. converting from the base C++ code to Lua. (this is actually a very good and game mod specific idea!)

**Respondent 9:**

* Expert programmer (7/7)
* Amateur Lua programmer (3/7)
* Games: Factorio
* Believes Lua is suitable.
* Feels that Lua sometimes feels like just a method of interacting and manipulating a series of tables.
* *The copy semantics are not opaque.* – I think they mean that the reference is not broken when creating a new object and assigning it to a current one.
* Finds the hardest part of mod development to be figuring out the API.

**Respondent 10:**

* Amateur programmer (3/7)
* Intermediate Lua programmer (4/7)
* Games: World of Warcraft
* Thinks Lua is great, but the API is not.

**Respondent 11:**

* Advanced programmer (5/7)
* Advanced Lua programmer (5/7)
* Games: World of Warcraft
* Thinks Lua is suitable for game modification development.
* Likes the simplistic syntax.
* Would like compilation warnings (since Lua doesn’t compile, this isn’t really possible) and proper typing (very possible).

**Respondent 12:**

* Advanced programmer (6/7)
* Intermediate Lua programmer (4/7)
* Games: World of Warcraft
* Thinks Lua is fine for game modification development.
* Doesn’t like using then and do after each structure (and presumably end).
* Believes other languages have better basic maths and table/array/object method syntactic sugar.
* Also wants better API documentation.
* Is a professional programmer, but mainly works on the front end.

**Respondent 13:**

* Advanced programmer (5/7)
* Advanced Lua programmer (5/7)
* Games: Civilization VI, Civilization V
* Believes Lua does its part in the *modification architecture* (Civ uses SQL, XML and C++ for modding too, but Lua is of course the primary scripting language) well.
* Problems arise from the API that Firaxis (the game’s developer) provides.
* Wants a method to find a value in a table (some kind of search function?)

**Respondent 14:**

* Advanced programmer (5/7)
* Advanced Lua programmer (6/7)
* Games: Civilization V
* Thinks that Lua is suitable for the task.
* Finds sharing information across contexts/files is a headache, but believes this is due to the game itself (although I disagree here).
* The Lua standard library provides nice basic functionality.
* Wants continue statements and switch statements.
* Wants more debugging tools and maybe even the ability to make test cases.

**Summary:**

* Programmer skill level breakdown:
  + Beginner (1): 0
  + Amateur (2, 3): 3
  + Intermediate (4): 1
  + Advanced (5, 6): 8
  + Expert (7): 2
* Lua programmer skill level breakdown:
  + Beginner (1): 0
  + Amateur (2, 3): 1
  + Intermediate (4): 2
  + Advanced (5, 6): 9
  + Expert (7): 2
* Games breakdown (was not posted to Roblox or Elder Scrolls Online communities):
  + World of Warcraft: 4
  + Civilization VI: 2
  + Civilization V: 3
  + Garry’s Mod: 2
  + Factorio: 2
  + Binding of Isaac: 3
  + One Step from Eden: 1
* Key points:
  + All respondents think that Lua is fit for purpose.
  + 9/14 (64.3%) of participants mention problems with the API of their respective games. Only those two develop mods for Garry’s Mod did not indicate this was a major issue.
  + 3/14 (21.4%) of respondents reported beginning programming by making mods in Lua.
  + The general consensus from respondents seems to be that they like the simplicity of Lua’s syntax.
* Ideas:
  + Some kind of in-line documentation with the game’s API (this could however be too big a task).
  + The continue keyword.
  + switch statements.
  + Compound assignment operators (+=, -=, \*=, /=, %=).
  + Increment/decrement operators (i++, i--, ++i, --i).
  + Additional IO support (refer to respondent 7).
  + Multithreading support (probably not feasible).
  + Optional typing/type enforcement.
  + map function (and we can expand this to other functional methods like filter and foldr).
  + Some kind of search functionality.
  + More debugging tools.
  + Something akin to a.length variable/function as opposed to the current # syntactic sugar.