

Medieval Millennium Events

The Lua code included in Medieval Millennium was written and revised many times over a period of several years. Early portions of the code were written when I was first learning the Lua language, and weren't necessarily rewritten later when my understanding had increased (as long as they were working correctly). As a result, there are sure to be inconsistencies in style, approach, and quality across different functions and files. Although a great deal of testing has taken place to try and eliminate as many bugs as possible, no guarantees can be made that every defect or deficiency has been found and fixed.

Lua events in Medieval Millennium are organized into numerous modules, each of which revolves around a distinct set of logical concepts. This usage of modules allows the code within Events.lua to remain more streamlined. These modules begin with the scenario initials "mm" and this reflects a key point: they are *scenario-specific*, containing code that is tailored to the exact needs and goals of Medieval Millennium. It would be possible to include some of the same features in a completely different and unrelated scenario, but this would involve copying and then editing the code in the module to adapt it to the new scenario. These are not intended to be generic modules that can be referenced seamlessly in any project. As a general rule, though, each module operates *independently* of the others to the greatest possible degree. With the exception of mmAliases.lua, there are no direct references from one module to another; any relationship management between them is brokered by the top-level Events.lua file.

Medieval Millennium also makes use of multiple function libraries that were developed alongside the project. These libraries provide a wide array of support functions that make scenario-specific events much more readable, less cumbersome to write, and easier to troubleshoot. The majority of these libraries are organized around the main Civ "objects" that were made available through TOTPP and Lua. In contrast to the modules referenced above, the code in these libraries is *scenario-independent* — that is, they could potentially be copied into (or referenced from) another TOTPP/Lua project and used without any edits. However, there may be desirable functions or features which are *missing*, and which could be useful in other scenarios, but which haven't been included yet simply because they weren't required for this project. These libraries are therefore not "finished products" and I may release updated versions of them, with enhanced functionality, at some point in the future — either independently, as part of an update to Medieval Millennium, or as part of another project entirely. (For example, this first version of utilTile.lua only supports calculations on flat maps, not those that wrap around, and is missing several functions that are likely to be useful in other projects.) The list of these libraries is:

- globalFunctions.lua
- log.lua
- utilCity.lua
- utilCivUI.lua
- utilImprovement.lua
- utilTech.lua
- utilTile.lua
- utilTribe.lua (note: would also include functionality for "leader" and "spaceship" objects, if necessary)
- utilUnit.lua (note: includes functionality for both "unit" and "unittype" objects)
- utilWonder.lua

On the Civfanatics.com forums, other Lua scenarios or separate modules have been released which overlap with portions of the functionality included in Medieval Millennium. Where I have borrowed from the work of others in the Civ community, or from other online sources, I've made every effort to assign credit and provide an appropriate reference within the code itself. However, there are also cases in which similar functionality was developed independently — typically because the code for Medieval Millennium had already been written but not yet released to the public. The fact that the Lua code within Medieval Millennium contains a unique set of libraries and modules, instead of utilizing ones already present in the community, is not intended as any type of negative statement regarding those. Rather, this is simply a byproduct of an independent and very lengthy development process. I sincerely appreciate and support the efforts of others (most notably Prof. Garfield) to make Lua event functionality accessible to more scenario developers.

If you are developing a Civilization II scenario utilizing Lua events and would like to copy, adapt, or incorporate any of the code released with Medieval Millennium into your own project, you are welcome to do so. Just as most scenario designers do their best to provide credits for icons and artwork that they borrow from other sources, I would encourage you to follow a similar practice regarding Lua code and reference this project as your source.