

BeardLib

Back in January of this year, we launched '[BeardLib](#)'. As you may have noticed it stopped receiving updates not so long after release. Now we are ready to show what we have been working on! and it is extremely exciting. The original version just came with some nice features for manipulating script data, which has managed to kick start some cool heist modifications. Which is something everyone has been pushing for since day 1 with modding. Well, today is the day that everything gets a whole lot better for custom heists and modding as a whole!

As you have seen we have finally released ViciousWalrus' custom heist, which was teased many weeks ago. This is the first fully custom heist, which was made using the [BeardLib Editor](#). The Editor is very much still in development so issues are expected (please post any bugs in the issues section on github). For those of you who were using the Script-Data converter and the Environment Editor those are now in the BeardLib Editor, alongside the Level Editor.

When we were developing the new version of BeardLib one of the main goals was to implement a new '[Module](#)' system. The goal of which would be to unify basic code for mods and thus lowering the barrier for users to create quality BLT mods. Similar to how BLT revolutionized Lua modding for Payday, allowing for a widespread increase in good Lua mods which were incredibly scarce before. With this modders will no longer have to implement basic features like Settings, Option menus or even rudimentary things like Localization or Class loading. The latter may only require a few lines, but their implementation as modules means that only an xml definition is required to achieve a fully functioning mod.

Probably the most important module that we currently have is the '[OptionModule](#)', this allows for a set of settings to be defined which will be automatically loaded and saved. This module also supports the ability to **automatically generate option menus**, with all the necessary features that are normally needed for an Option menu. It is at this point that I would like to talk about PD:TH Hud, as you have seen, it contains a fancy new Portrait menu, new icons and some other minor changes. Which is nice, but that's not why it's here, it is the primary example for the implementation of the modules ([see xml config](#)). It was the primary point of development of the module system, so it uses the system in its purest form. As you can see in the xml config, it defines an Options module with its defined settings. This is all that is required to implement an Options menu, which is just as good, if not better than a generic, manually written one and it is all automatically generated.

PD:TH Hud does use the purest form of the module system that you can do currently, however this does require some modification to how most mods are created currently. If you would like to see a semi-pure implementation (see [editor](#)), this allows for utilizing the module system without requiring a substantial amount of modification to the existing mod. We also have a template for a pure BeardLib mod [here](#)! This can be used as a basis of creating your own mod with BeardLib.

We've also created a Map Framework, which is powering Out of Frame. This Framework allows for the automatic addition of maps to the game, instead of having to copy over a BLT mod and editing it to your needs for adding your heist. This framework will do all of that for you, all you have to do is just define information about your Narrative(s) and Level(s) in an xml file and you're all set! It even handles loading of packages. These Map XML's can also utilize all of the existing BeardLib modules and vice-versa! This allows maps to utilize exclusive hooks, which will be run when the level is being played.