Introduction

FabIO is a C++ Discord bot using Sleepy Discord (docs at: https://yourwaifu.dev/sleepy-discord/documentation.html). Internally it hosts a LUA state for game programming, and renders flat html files for visuals. It is meant as a step towards the Zero Project, and allows a wide variety of rpg and board-like games to be played via Discord chat.

Given you have access to this forum, you can program the bot using the guide below. Essentially it is as simple as making a wiki page for your game, and then telling the bot to load the code from that wiki page.

Programming Details

I am not going to list all the programming details here for the bot, but the general overview. If you are working on these scripts I'm sure you'll be in Discord to ask any specific questions you might have. Also, I'm recording a bit here to make sure I remember how things work!

Basic Concepts

The FabIO bot runs LUA scripts at its core. It is just a frame for them, and allows those scripts to be called from Discord and then output an html view that is fed to a local server. In this way the scripts are the 'magic' between Discord and that webpage. To organize this all in a useful way, the bot supports: games and instances.

- **Games**: These are groups of scripts that are all loaded to support the functions of a game. They also contain embedded themes for the web view, but those can be overridden by an instance easily. So a game is just a bunch of scripts, that give functions to the players. Game's will be able to inherit from other games, so it's possible your game 'D20' might depend on 'std-lite' and just add scripts on top of the ones already provided by game you are deriving from. Games support both multiple and hierarchical inheritance, meaning you can both inherit from multiple games, and those games can inherit from other games. Each game itself is a page of code on this wiki, as seen below: Game Code Pages.
- **Instances**: These are specific instances of data for a game in use. An instance is tied to the user that created it, but not actually to the game itself (it's just data). Special theme data in instances overrides the games internal theme information so it can be skinned specifically for a given game, giving the final html output it's own look.

Wiki Linkage

The FabIO bot maintains local copies of scripts from games on this wiki. Editing a game page here has no immediate effect. You must tell the bot in DM to $upload <game-name>. Then it'll parse the wiki page, find all the scripts, copy them locally, and compile them into a game state for use. Any errors it finds will be reported directly to you in response.

HTML Output

You can find the output of the bot here: http://fabio.unusualperson.com/games.

Game Code Pages

- boot - This is a common script loaded by all games before anything else starts. Maintained by JasonP.
- std-lite - A generic all purpose game with support for a lot of very basic stuff. Counters, Dice, Sheets, etc. Maintained by JasonP.

From:
https://wiki.wishray.com/ - **Wishray Wiki**

Permanent link:
https://wiki.wishray.com/doku.php?id=fabio:fabio

Last update: **2020/03/29 08:08**