MOTI BARSKI

AUXILIARY MODULES for the LIVINGRIMOIRE

preface

utilizing auxiliary modules for your living grimoire skills is in no way mandatory.
these are simply modules that ease writing skills and make the waifubot more human.
this is based on experiments with a variety of skills I have written.

the auxiliary directories (respective to the programming languages) will be updated when needed.
you can write and post your own auxiliary modules/classes just like the livingrimoire skills

HTTPS://JIZZ.IS/FORUMS/THE-LIVINGRIMOIRE.3/

algorithm dispensers :

these are classes that can return an algorithm
their internal logic can shuffle, cycle algorithms or change algorithms according to
moods or status , to return the next time
(you can also use your own custom logic)

Defcon Detectors :

these classes detect key inputs, for example goals, dangers or items.

learnabillty modules : loggers :

a script is a collection of narratives sorted in some way :
by defcons
by subject/category
with fillers

there are 3 main parts to a script :

the narrative :
1.1 a mold (see perchance chatbot alg (DiJoker skill is a fine example for it))
1.2 parameters

2.1 the narrative mixer which generates paragraphs or chapters out of narratives
2.2 narrative emotional context which sets which molds and params will be used
the use of cuss words for example.

params can be sorted by :

learned by experience
oldest
newest
untested/suggested (stories heard)
random (imagined)

the purpose of loggers is to convey params to crafters and map info to maps

learnability modules : crafters :

these turn a logger narrative into an algorithm

cooking recipes into actual cooking for example

learnability modules : spider sense :

a flag event is an event that predicts the occurrence of a key event (goal or defcon),
key events are defined in the defcon detector

learnability modules : input filters :

```
/*
 * the input filter only lets data relevant to key events (defcons) through it
 * can be status changes in weather, sentences that contain certain inputs, data
 * at a particular space of time for example
 */
```

paradoxers :

upon detection of a flag event the spider sense should raise a paradoxing action
which means an action that would negate the occurrence of the defcon event before it happens

for example : when a police officer tells someone to put the weapon down

if, however the event is a goal, an anticipation action should be raised

Learnability :

```
/*
 * this class can facilitate checking if a goal manifested or if defcons
 * manifested. therefore as a boolean gate it can be used to trigger
mutations
 * of Alg Parts or even used externally (within a skill) see also the ML
search
 * algorithm
 */
```

map :

this stores key inputs with it's respected coordinates
a compass should be used to detect that the bot is moving in order for data such
as from watching a TV show will not register as map points.

responders :

these simply return a string reply. be it warnings, threats, conversation engagement encouragements
and questions.

trigger classes :

these are custom boolean triggers.
they all must extend the TrGEV3 class so if needed you can combine multiple triggers using an
EV3DaisyChain object.

the trigger classes have a naming convention : their name should start with Trg.

there are 4 main types of trg objects. based on the DiParrot skill, these are the triggers responsible
to make a waifubot feel alive to the user :

EVENT(TIME TRIGGERED) :

for example :
at 5 o'clock

REPLY :

triggered by certain input

STAND BY :

accumulated input triggered

for example :
trigger after x minutes of no input
or trigger after being cussed at for y times
or after getting patted so and so times by the user

TOGGLERS :

these are basically like mood swings
the gates logic varies and will react(return false or true) differently based on prev input

FRIEND :

these can be considered a toggler triggers
it checks if a friend is present. and so the behavior or actions can very to that context, for example, be more talkative.

aux misc :

this dir contains a misc of helpful auxiliary modules.

utils :

these modules sum up and ease access to common complex tasks

TO SUM IT UP :

TRIGGERS -> LEARNABILITY MODULE -> ALG DISPENSE
NO TRIGGER -> LEARNABILITY MODULE

tho you can obviously mix up the modules like :
trigger->responder->alg
waifubot gets petted x times(standby trigger)-> than starts a new type of purr (like a cat)

mostly learnability modules use :
input filters
Defcon Detectors

and ***mostly*** algorithms use the modules:
responder
map

ChobitLight

this new class is an alternative to the ChobitV2 class
the ChobitsLight classes c'tor uses the PersonalityLight class instead of the Personality class.

the ChobitsLight class does not have default permissions or names

so the
-permission
-personality

classes are not used (not by default by the ChobitsLight class).

this way, each class can handle it's own permissions within the friend trigger/module (for example),

which enables the use of fast short algorithms rather than placing all the logic into AP classes

AP = ALGORITHM PART (SEE LIVING GRIMOIRE BOOK)

check the GitHub or forum links once in a while for updates and member posts :

https://github.com/yotamarker/public-livingrimoire

the auxiliary modules can be found in the "livingrimoire start here" directory per programming language

https://jizz.is/

AUXILIARY MODULES for the LIVINGRIMOIRE

Auxiliary modules for, Living Grimoire (artificial general intelligence software design pattern), skills.

This book is intended for battle programmers who wish to write better skills, and do so in a more efficient way.

You will also learn about the new ChobitsLight class, which is an alternate version of the ChobitsV2 class.