

MOTI BARSKI



AUXILIARY MODULES for the
LINGRIMOIRE

preface	4
algorithm dispensers	4
Crafters	4
Defcon Detectors	4
Learnability	5
Input filters	5
Map	5
Responders	5
Triggers	6
Misc	6
Upper Chobits	6
Lower Chobits	6
to sum it up	7
Codex	7
LGCore auxiliary modules	7
AlgDispenser	8
AXLearnability	8
Catche	8
CombinatoricalUtils	8
Cycler	8
DrawRnd	8
EmoDetectorCurious	9
EmoDetectorHappy	9
EmoDetectorStressed	9
EV3DaisyChainAndMode	9
EV3DaisyChainOrMode	9
ForcedLearn	9
InputFilter	9
LGTypeConverter	9



Map	10
RepeatedElements	10
Responder	10
SkillHubAlgDispenser	10
SpiderSense	10
TrGEV3	10
TrgMinute	11
TrgParrot	11
TrgSnooze	11
TrgTime	11
TrgTolerance	11
UniqueItemSizeLimitedPriorityQueue	11
UniqueItemsPriorityQue	12

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 livinggrimoire skills

[HTTPS://WWW.YOTAMARKER.COM/F2-THE-LIVINGGRIMOIRE](https://www.yotamarker.com/f2-the-livinggrimoire)

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). automatic programing may also be included to this category.

CRAFTERS

this transform strings into AlgParts or entire Algorithm Objects

DEFCON DETECTORS

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

LEARNABILITY

this are beyond omega level modules, because they enable learning. prediction of events or a decision to mutate behavior based on experience deduction may also be included with the learnability category .

INPUT FILTERS

the input filter only lets data relevant to key events through. it can be status changes in weather, sentences that contain certain inputs, defcons, colors, shapes and so on.

MAP

this stores key inputs with its 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.



TRIGGERS

these are advanced Boolean gates. their name should start with Trg. they signal events or robust conditions. a simple example would be a time stamp or a countdown gate.

the trigger category can be further divided to sub categories:

events (time)

reply (to certain inputs)

stand by (no input or output for a certain while)

togglers : mood swings or hardware states (temperature , battery level)

friend: is friend present? (this module is basically a context)

MISL

miscellaneous auxiliary modules, see codex for some examples.

these modules sum up and ease access to common complex tasks.

you can end their name with Util, for example RegexUtil.

UPPER CHOBITS

modules for skills for Chobit Objects responsible for morphing reality or initial input. for example, pain killers filter out pain inputs.

LOWER CHOBITS

modules for skills for Chobit Objects responsible for morphing outputs, for example adding the ending nya to every output sentence, or filtering out no no words.

TO SUM IT UP

triggers -> learnability module -> alg dispense
no trigger -> learnability module

then you can obviously mix up the modules like :
trigger->responder->alg

waifubot gets petted x times(standby trigger)-> then 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.

[HTTPS://GITHUB.COM/YOTAMARKER/
PUBLICLIVINGRIMOIRE](https://github.com/YOTAMARKER/publiclivingrimoire)

Codex

LGCORE AUXILIARY MODULES

LGFIFO : a first in first out queue.

PlayGround : class to manage system time elements.

RegexUtil : a class to ease working with regular expressions.

DeepCopier : creates a deep copy for lists.

TimeGate : a Boolean gate that closes automatically after a set amount of time.



ALGDISPENSER

category: algorithm dispenser

description:

outputs an algorithm out of a selection of algorithms

AXLEARNABILITY

category: learnability

description:

recommends mutations to the active algorithm used for accomplishing a goal based on positive or negative(defcon) inputs,

providing tolerance for mistake that prevents algorithm mutation
(the more tolerance the less chance for algorithm mutation, persistence in other words)

CATCHE

category: misc

description:

this is simply a dictionary Object with limited size.

COMBINATORICALUTILS

category: misc

description:

takes in lists of Strings and returns all permutations (combos) between said lists

(via the result arrayList Object)

CYCLER

category: misc

description:

counts down, then resets the count to the initial limit(maximum)

meaning: 3,2,1,0,3,2,1,0

DRAWRND

category: misc

description:

draw a random element, than removes said element out



so every element is used only once, like a deck of cards

EMODETECTORCURIOUS
EMODETECTORHAPPY
EMODETECTORSTRESSED

category: emotion detectors

description: detect respective emotions

EV3DAISYCHAINANDMODE

category: trigger

description:

this class connects several logic gates triggers together
using and logic

EV3DAISYCHAINORMODE

category: trigger

description:

this class connects several logic gates triggers together
using or logic

FORCEDLEARN

category: responder

description:

remembers key inputs because they start with keyword
also can dispense said key inputs

example: say hello

saves the word hello among other inputs

INPUTFILTER

category: InputFilter

description:

filter out non-relevant input
or filter in relevant data

LGTYPEDCONVERTER

category: misc

description:

converts String types to other Object types.

yes the programming language has code to do that anyways, but this is
simpler



MAP

category: map

description:

a map with points and location descriptions

REPEATEDELEMENTS

category: defconDetectors

description:

detects repeating elements in a feed of input
with respect to the feeds length:

1,2,3,4,1 : 1 repeated in a feed length of 5

RESPONDER

category: Responder

description:

simple random response dispenser

SKILLHUBALGDISPENSER

category: algorithm dispenser

description:

super class to output an algorithm out of a selection of skills
engage the hub with dispenseAlg and return the value to outAlg attribute
of the containing skill (which houses the skill hub)
this module enables using a selection of 1 skill for triggers instead of having
the triggers engage on multiple skill

the method is ideal for learnability and behavioral modifications

use a learnability auxiliary module as a condition to run an active skill shuffle
or change method

(rndAlg , cycleAlg)

moods can be used for specific cases to change behavior of the AGI, for
example low energy state

for that use (moodAlg)

SPIDERSENSE

category: learnability

description:

spider sense is tingling? event predicted?



TRGE V3

category: trigger

description:
super class for trigger boolean gates
the trigger gates do not have to extend this class

TRGMINUTE

category: trigger
description:
trigger true at minute once per hour

TRGPARROT

category: trigger
description:
simulates a parrot chirp trigger mechanism
as such this trigger is off at night
in essence this trigger says: I am here, are you here? good.

TRGSNOOZE

category: trigger
description:
this boolean gate will return true per minute interval
max repeats times.

TRGTIME

category: trigger
description:
this Boolean gate returns true once a day during the set time

TRGTOLERANCE

category: trigger
description:
this boolean gate will return true till depletion or reset()
it is a count down Boolean gate

UNIQUEITEMSIZELIMITEDPRIORITYQUEUE

category: misc
description:
a UniqueItemsPriorityQue with a limited size



UNIQUEITEMSPRIORITYQUEUE

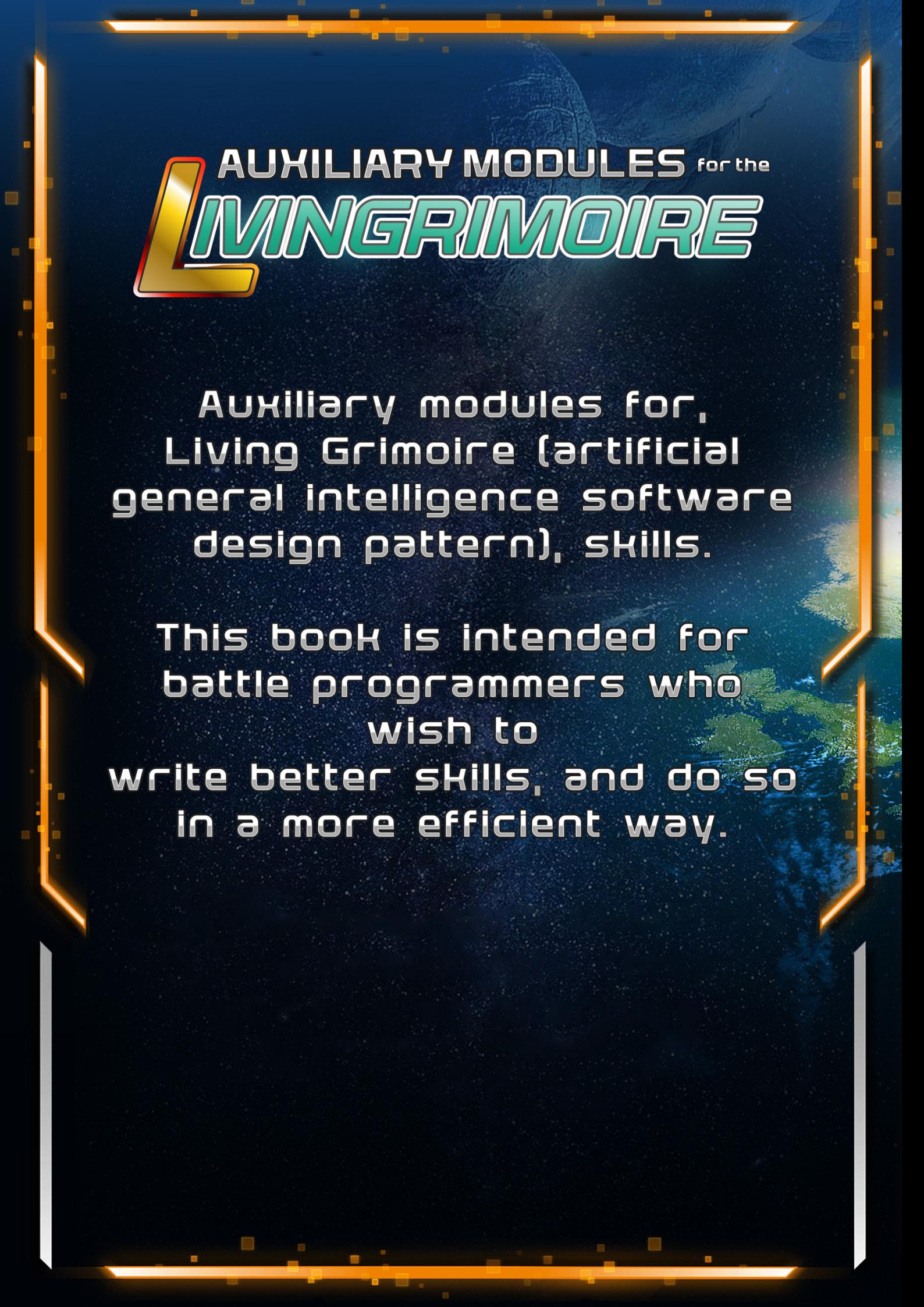
category: misc

description:

a priority queue (LIFO first in first out) where no item repeats
pulling out an Object will return the earliest Object that was inserted
and remove it from the queue







AUXILIARY MODULES for the ***LIVING GRIMOIRE***

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.