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.

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://www.yotamarker.com/f2-the-livingrimoire

SUGGESTED SKILL SKELETONS

triggers/cron job -> learnability module -> alg dispense
no trigger -> learnability module

tho you can obviously mix up the modules like :
trigger/cron job>responder->alg

mostly algorithms use convo(conversational) ,hub, and hardware modules

https://github.com/yotamarker/publiclivingrimoire

AXModules

PDF made by "Create Dictionary"
נובמבר, 2023 - 06:52 אחה"צ 23

Alerter *(Secreterial)*:

Manages reminders

Next reminder; Clear reminders; Remind me to eat at 11:30; Say "thank you" after reminder or it will be auto deleted

AlgDispenser *(Hub)*:

Output alg from list; Active alg can be randomized or cycled(next alg). The class has alg CRUD methods

AnnoyedQue :

Learns inputs containing strings that are repeatedly used by others(ear param)

AXCmdBreaker *(Trigger)*:

Separate command param from the command

Say hello, will return hello as param, assuming say is the command (conjuration param)

AXContextCmd *(Trigger)*:

Engages commands and context commands. The command gives meaning to the context command ; Context is active while commands or other context commands are issued. Unrelated input will undo the context till the next context cmd is issued

What is the time can be a context command. what is it, is a command, that will only engage after a command was issued.

AXFriend *(Misc)*:

Registers friend. Requests friend; Reset() // reset to friendless

I am chi: bots offers friendship; I am shuki: bot registers friend if it has non.

AXGamification *(Learnability)*:

Gamification module. Counter represents grind, and can be decreased when requesting a reward. Like in the app habitica. Rewards can be skill activations, abilities, personality traits, and even punishments.

AXInputWaiter *(Misc)*:

Returns true x times or till any input (string input)

AXKeyValuePair *(Object extension)*:

Key value object, like the ones inside a dictionary

AXLearnability *(Learnability)*:

returns true to recommend behavior modification. Triggered by x failures after alg deployment. Resets said x on goal manifestation.; Note the module does not suggest a muttation solely on the activation of an alg, because this would be like needing a validational "thank you" per goal, which takes the fun out.

When mutateAlg() returns true, mutate the default alg used by the skill; PendAlg() use this when deploying an alg. It has a bigger chance of mutating an algorithm because it counts utilizing an algorithm as a negative. Too many tries mean the alg is not that reliable.; Use pendAlgWithoutConfirmation() when deploying an alg

AXLHousing *(Convo)*:

Superclass for string decorations

Hello -> hello nyaa

AXLHub *(Convo, Hub)*:

Hub of AXLHousing (string decorators)

AXLMorseCode *(Convo)*:

Comvert to morse code

AXLMorseDecoder *(Convo)*:

Decodes morse code

AXLNeuroSama *(Convo)*:

Adds heart, wink or nothing to a string

AXLSpeechModifier *(Convo)*:

Scans a string verbatim and replaces certain words(keys) to values in a dictionary.; Can be used to add soeech styles or censor words

AXMachineCode *(Misc)*:

A dictionary with a default return value. Used to ease machine code

AXNightRider *(Hardware)*:

Nightrider display simulation for led lights.

AXNPC *(Convo)*:

Has N percent chance to return a response. Use this module to output strings that are not dependent on context, such as pet talk.; A limited sized que holds the responses, and new responses push out underuse responses. ; Use responsePlus() to increase chance of reply. Recommended if any input has been detected.; Has a forced output method with 100% chance of response output

Has method to learn new responses: "say hello there": say is a wild card and can be modified. The parameter hello there will be added to the response que.

AXNPC2 :

A subclass of AXNPC, can also learn inputs containing strings that are repeatedly used by others(via ear param)

AXPassword *(Misc)*:

A password boolean gate; Suggested algorithm: at a set time have the bot change the password code and reveal the new code. Once per month or once per week.; This is a boolean gate which requires a password to open.

CodeUpdate("code 1234") set new pass // while gate is open; OpenGate("code 1234") open the boolean gate

AXPrompt :

Prompts the user for input. Only preset regex per prompt is accepted in order to progress to the next prompt. Results(user answers) are extracted as a keyValuePair type via a method.

See class documentation for exaple use of this module.

AXStrategy *(Learnability)*:

Outputs strategy per input as context. This can be used for fighting or gaming. Use the evolve method to change which strategies are active.

Exaple context: defense, attack, grab

AXStringSplit :

May be used to prepare data before saving or after loading. The advantage is having less data fields. For example: skills:s1_s2_s3

AXStrOrDefault :

Returns the string param unless it's empty, in which case the default param is returned

AXTimeContextResponder *(Convo)*:

Responds in context to the time of day (morning afternoon evening and night).; The class has a responder object respective to part of day. 4 in total.

buttonEngager *(Hardware)*:

Simulates a btn press. Teturns true only once per press. The algorithm is also useful for Arduino.

Catche *(Object extension)*:

Limited sized dictionary

ChatBot *(Convo)*:

Chatbot module with mutatable parameters. Can be used for advanced conversation skills, script generation and even visual novels. When params are added old params are forgotten. ; See class documentation for usage example

CombinatoricalUtils *(Misc)*:

Returns all combos for varargs string lists

Cron *(Cron job)*:

Triggers true, limit times, after initial time, and for every minutes interval. The counter resets at initial time, assuming trigger method was run.; Cron jobs

Cycler *(Counter)*:

Counts down from limit to 0 to Count and so on.

Differ *(Misc)*:

Calculates difference between prev and current state. Used for battery level as bot sense of hunger

DiSysOut *(Hardware)*:

Example hardware skill. Varies depending on the IDE.; Prints output

DrawRnd *(Misc, Convo)*:

Draws a random element and removes said element. Much like a deck of cards

reset() reset and refill all removed elements in the object.

EmoDetectors *(Trigger)*:

Detect preset words, pointing to a specific emotion

EV3DaisyChain *(Trigger)*:

Chains trigger gates together with and or or.; Under use

ForcedLearn *(Learnability)*:

Saves words when commanded

Say hello. GetRandomElement may return the word hello

InputFilter *(Misc)*:

Filters out non relevant input

LGTypeConverter *(Misc)*:

Converts string to int or double. The advantage of this class is you don't need to search for the conversion code for the 1000th time

Map *(Misc)*:

Map object. A dictionary representing a 2x2 matrix map and location description. Where the bot sleeps is considered 0,0

NumToWord *(Convo)*:

Converts a number to words

123->one hundred twenty three

OutputDripper *(Convo, Misc)*:

Drips true once every limit times; Shushes the AI enough time for the user to reply

PercentDripper *(Misc, Trigger)*:

has an N chance of returning true. N can be set permanently or temporarily with a method

PersistentQuestion *(Learnability)*:

Asks a question several times in various ways or till it gets an answer.

PersistantQuestion persistantQuestion = new PersistantQuestion(); persistantQuestion.addPath("yes",new DrawRnd("I love you", "do you love me?","please do you love me","you love me don't you ?")); persistantQuestion.addPath("no",new DrawRnd("you're annoying", "I'm leaving","good bye","you love me don't you ?")); persistantQuestion.activate(); for (int i = 0; i < 10; i++) { System.out.println(persistantQuestion.process("")); } persistantQuestion.activate(); System.out.println(persistantQuestion.process("")); System.out.println("answering no:"); System.out.println(persistantQuestion.process("no")); persistantQuestion.activate(); for (int i = 0; i < 10; i++) { System.out.println(persistantQuestion.process(""));; log() unlike process() will save any answer, while process will only save preset answers(answer param in addpath method)

RefreshQ :

Subclass of uniquesizelimitedq. This is a FIFO Q. If a contained item is inserted, it moves to the back of the q delaying it's poll priority, thus refreshing the item in memory.

Responder *(Convo)*:

Returns a random word out of a list of words.

Responder1Word *(Trigger, Learnability, Convo)*:

Returns a random word out of a word list. Only accepts single words. Forgets old words when learning new words.

SkillHubAlgDispenser *(Hub)*:

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 multible skill. the methode is ideal for learnability and behavioral modifications. use a learnability auxiliary module as a condition to run an active skill shuffle or change methode (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 *(Learnability)*:

Event prediction; Can be used for warnings and cognition training , as in Pavlov's bell experiment

Learn method to learn input, getSpiderSense is true if event string is predicted.

Timeaccumulator *(Trigger)*:

A counter that increments as a result of time. The tick is changeable

ToDoListManager *(Secreterial)*:

Saves and outputs tasks.; Forgets mentioned tasks; Outputs old tasks

TrgArgue :

Retorts to thank yous; Retorts to arguments; Counts persistence of arguments or thank yous; Req: tell me the time, persistence: (contains)please/would you: as in "please tell me the time"

Unrelated input resets the counter; EngageCommand(str) returns 0 for no engagement, 1 for request, 2 for persistence (argument or thank you)

TrgEveryNMinutes *(Cron job)*:

Trigger returns true every minutes interval, post start time

TrgMinute *(Cron job)*:

Trigger method returns true at minute once per hour; Hourly Cron job

TrgSnooze *(Trigger)*:

This boolean gate will return true per minute interval, max repeats times.; Somewhat like the Cron class

TrgTime *(Trigger)*:

Returns true once per 24 hours at time stamp

11:44 is an example of a time stamp

TrgTolerance *(Trigger, Counter)*:

This boolean gate will return true till depletion.

UnigueItemPriorityQue *(Misc)*:

A LIFO que with unique items

UniqueItemSizeLimitedQue *(Misc, Object extension)*:

LIFO Que with limited size AND ONLY UNIQUE NON REPEATING items

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.